

DIAGNOSIS PENYAKIT TANAMAN CABAI MENGGUNAKAN METODE *MODIFIED K-NEAREST NEIGHBOR* (MKNN)

(Studi Kasus: BPTP Karang Ploso Malang)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

HADI DWI ABDULLAH HAMID

NIM: 125150207111053



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

DIAGNOSIS PENYAKIT TANAMAN CABAI MENGGUNAKAN
METODE *MODIFIED K-NEAREST NEIGHBOR* (MKNN)
(Studi Kasus: BPTP Karang Ploso Malang)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Hadi Dwi Abdullah Hamid

NIM: 125150207111053

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Nurul Hidayat, S.Pd, M.Sc

NIP. 19680430 200212 1 001

Dr. Ratih Kartika Dewi, S.T., M.Kom

NIK. 201503 890520 2 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astuti Rahmawati, S.T, M.T, Ph.D

NIP. 197710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Desember 2018



Hadi Dwi Abdullah Hamid

NIM: 125150207111053

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)”, yang diajukan untuk menempuh Ujian Akhir Sarjana Program Strata Satu Jurusan Informatika.

Selesainya penulisan skripsi ini tidak terlepas dari peran serta berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala limpahan karunia, rahmat, dan hidayah-Nya.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
5. Bapak Nurul Hidayat, S.Pd, M.Sc dan Ibu Ratih Kartika Dewi, S.T., M.Kom, selaku dosen pembimbing penulis yang dengan sabar memberikan kritik, saran, serta arahan yang baik dalam proses pengerjaan skripsi ini.
6. Seluruh Bapak dan Ibu Dosen serta Staff Akademik di Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang.
7. Bapak dan Ibu, Saudara serta Keluarga Besar atas segenap dukungan dan kasih sayang yang telah diberikan kepada saya.
8. Siti Rima Julianti, S.Pi.. sahabat saya GP Family, PI Malang, generasi *alpha* FILKOM UB, Korlap 2014 Universitas Brawijaya serta sahabat yang terbatas untuk dicantumkan karena banyaknya teman yang peduli dan mendukung pada masa kuliah sampai pengerjaan Skripsi ini selesai.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak terdapat kekurangan. Untuk itu penulis menyampaikan permohonan maaf sebelumnya, serta sangat diharapkan kritik dan saran yang bersifat membangun dalam penyempurnaan di masa mendatang.

Malang, 12 Desember 2018

Penulis
Hadidwiah@gmail.com

ABSTRAK

Hadi Dwi Abdullah Hamid. 2018. Diagnosis Penyakit Tanaman Cabai Menggunakan Metode Modified K-Nearest Neighbor (MKNN). Skripsi Program Studi Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Nurul Hidayat, S.Pd, M.Sc dan Ratih Kartika Dewi, S.T., M.Kom.

Cabai merah merupakan salah satu jenis sayuran yang cukup penting di Indonesia, baik sebagai komoditas konsumsi di dalam negeri maupun sebagai ekspor. Cabai merah memiliki nilai gizi tinggi, juga mempunyai nilai ekonomi tinggi. Namun produktivitas cabai merah nasional rendah yaitu 7,34 ton/ha, padahal potensinya mampu mencapai 12 ton/ha. Dalam suatu periode tanam, cabai bisa dipanen beberapa kali bila musim dan perawatannya baik dapat 15-17 kali, namun umumnya hanya 10-12 kali. Rendahnya produktivitas bisa disebabkan oleh berbagai faktor, yaitu mutu benih kurang baik, tingkat kesuburan tanah semakin menurun, teknik budidaya kurang baik, serta permasalahan hama dan penyakit tanaman. Untuk menangani hal tersebut diperlukan teknologi dengan menerapkan metode klasifikasi yaitu Modified K-Nearest Neighbor (MKNN). MKNN adalah pengembangan dari metode KNN yang dirancang untuk mengatasi kelemahan dari jarak data dengan weight pada KNN. Metode tersebut mempelajari berdasarkan 18 gejala penyakit dengan proses perhitungan jarak euclidean, perhitungan validitas dan perhitungan wighted voting yang menghasilkan penetapan kelas klasifikasi berdasarkan nilai K yang ditentukan. Hasil pengujian menggunakan K=5 mendapatkan akurasi sebesar 94%, kemudian K=8 akurasi sebesar 92%, K=11 akurasi sebesar 88% dan pengujian K=14 menghasilkan akurasi sebesar 88%. Berdasarkan hasil yang didapat, metode Modified K-Nearest Neighbor (MKNN) menunjukkan akurasi yang baik untuk melakukan klasifikasi penyakit cabai.

Kata Kunci: *Modified K-Nearest Neighbor (MKNN)*, Klasifikasi, Penyakit, Cabai.

ABSTRACT

Hadi Dwi Abdullah Hamid. 2018 . Diagnosis of Chili Crops Using the Modified K-Nearest Neighbor (MKNN) Method. Thesis of Informatics Study Program / Computer Science, Faculty of Computer Science, Brawijaya University. Supervisor: Nurul Hidayat, S.Pd, M.Sc and Ratih Kartika Dewi, S.T., M.Kom.

Red chili is one of the most important vegetables in Indonesia, whether it is as a commodity that is consumed domestically and as an export commodity. As vegetables, beside red chili has a high nutritional value, it also has a high economic value. However, the productivity of national red chilli is still very low at 7.34 tons / ha, whereas the actual yield can potentially reach 12 tons / ha. In a planting period, chili can be harvested several times. If the season and the treatment is very good, chili can be harvested 15-17 times but generally, it can be harvested only 10-12 times. The low productivity of chili can be caused by a variety of factors, including poor quality of chili seeds, decreasing soil fertility, bad implementation of cultivation techniques, plant pest and disease problems. In order to handle this, technology is needed by applying one of the classification methods, namely Modified K-Nearest Neighbor (MKNN). Modified K-Nearest Neighbor (MKNN) is the development of the KNN method that has been designed to overcome the weaknesses of the distance between data and weight in the KNN. The method analysis is based on 18 symptoms of the disease with the process of calculating euclidean distance, calculating the validity and calculation of wighted voting which result in the determination of the classification class based on the specified K value. The test results showed that when using the value $K = 5$ produces an accuracy of 94%, then $K = 8$ produces an accuracy of 92%, $K = 11$ produces an accuracy of 88% and testing $K = 14$ produces an accuracy of 88%. Based on the results obtained, the Modified K-Nearest Neighbor (MKNN) method showed good accuracy for classifying chili disease.

Keyword: Modified K-Nearest Neighbor (MKNN), Classification, Disease, Chili.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR SOURCE CODE	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Tanaman Cabai	6
2.3 <i>Data Mining</i>	7
2.4 Klasifikasi.....	9
2.5 <i>Modified K-Nearest Neighbor (MKNN)</i>	9
2.6 Pengujian Akurasi	12
BAB 3 METODOLOGI	13
3.1 Studi literatur	14
3.2 Pengumpulan Data	14
3.3 Analisis Kebutuhan	14
3.4 Pengolahan Data.....	15
3.5 Perancangan Sistem.....	15
3.6 Implementasi	17

3.7 Pengujian Sistem.....	17
3.8 Analisis	18
3.9 Kesimpulan.....	18
BAB 4 PERANCANGAN.....	19
4.1 Penyakit Tanaman Cabai.....	19
4.2 Perancangan Proses.....	27
4.2.1 Proses Klasifikasi <i>Modified K-Nearest Neighbor</i> (MKNN).....	28
4.2.2 Perhitungan Jarak Euclidean Antar Data Latih	30
4.2.3 Perhitungan Validitas	30
4.2.4 Perhitungan Jarak Euclidean Antara Data Latih dan Data Uji	32
4.2.5 Perhitungan <i>Weight Voting</i>	33
4.2.6 Penentuan Kelas Data Uji	34
4.3 Perhitungan Manual	35
4.4 Perancangan Antarmuka	39
4.4.1 Desain Tampilan Awal	39
4.4.2 Desain Tampilan Data Uji	40
4.4.3 Desain Proses MKNN	41
BAB 5 IMPLEMENTASI	42
5.1 Implementasi Sistem	42
5.1.1 Spesifikasi Perangkat Keras	42
5.1.2 Spesifikasi Perangkat Lunak.....	43
5.2 Batasan Implementasi	43
5.3 Implementasi Algoritma	43
5.3.1 Implementasi Perhitungan Jarak Euclidean Antar Data Latih	43
5.3.2 Implementasi Perhitungan Nilai Validitas	44
5.3.3 Implementasi Perhitungan Jarak Euclidean Data Latih dan Uji ..	46
5.3.4 Implementasi Perhitungan <i>Weight Voting</i>	47
5.3.5 Proses Klasifikasi	47
5.4 Implementasi Antarmuka	48
5.4.1 Implementasi Antarmuka Awal	49
5.4.2 Implementasi Antarmuka Data Uji	49
5.4.3 Implementasi Antarmuka Proses MK-NN.....	50

BAB 6 PENGUJIAN DAN ANALISIS.....	51
6.1 Pengujian Akurasi Sistem.....	51
6.1.1 Pengujian dengan nilai $K = 5$	54
6.1.2 Pengujian dengan nilai $K = 8$	57
6.1.3 Pengujian dengan nilai $K = 11$	60
6.1.4 Pengujian dengan nilai $K = 14$	62
6.2 Analisis	66
BAB 7 KESIMPULAN DAN SARAN	67
7.1 Kesimpulan.....	67
7.2 Saran	67
DAFTAR PUSTAKA.....	68
LAMPIRAN	69
A. Data Penyakit Cabai.....	69



DAFTAR TABEL

Tabel 4.1 Kode dan Gejala Penyakit Tanaman Cabai	27
Tabel 4.2 Kode dan Penyakit Tanaman Cabai	27
Tabel 4.3 Data Latih	35
Tabel 4.4 Data Uji	35
Tabel 4.5 Jarak Euclidian Antar Data Latih	36
Tabel 4.6 Euclidian antar data latih yang telah diurutkan	36
Tabel 4.7 Euclidian antar data latih sebanyak K	36
Tabel 4.8 Validitas Data Latih	37
Tabel 4.9 Euclidian Data latih dan Data Uji	37
Tabel 4.10 Weighted Voting	38
Tabel 4.11 Nilai Weighted Voting yang telah diurutkan	39
Tabel 5.1 Spesifikasi Perangkat Keras	42
Tabel 5.2 Spesifikasi Perangkat Lunak	43
Tabel 6.1 Data Uji 1	51
Tabel 6.2 Data Latih	53
Tabel 6.3 Hasil Uji K = 5	54
Tabel 6.4 Hasil Uji K = 8	57
Tabel 6.5 Data Uji K = 11	60
Tabel 6.6 Hasil Uji K = 14	63

DAFTAR GAMBAR

Gambar 2.1 Proses Penemuan Pengetahuan	8
Gambar 2.2 Proses Klasifikasi	9
Gambar 3.1 Alur Metode Penelitian	13
Gambar 3.2 Desain Perancangan Sistem	16
Gambar 3.3 Blok Diagram Proses Sistem	17
Gambar 3.4 Diagram Blok Pengujian Akurasi	18
Gambar 4.1 Busuk Buah	20
Gambar 4.2 Busuk Buah	20
Gambar 4.3 Busuk Buah	20
Gambar 4.4 Busuk Buah	21
Gambar 4.5 Bercak Daun	21
Gambar 4.6 Bercak daun	22
Gambar 4.7 Bercak Daun	22
Gambar 4.8 Layu Fusarium	23
Gambar 4.9 Layu Fusarium	23
Gambar 4.10 Layu Fusarium	23
Gambar 4.11 Layu Fusarium	24
Gambar 4.12 Layu Fusarium	24
Gambar 4.13 Layu Fusarium	24
Gambar 4.14 Virus Gemini	25
Gambar 4.15 Virus Gemini	25
Gambar 4.16 Virus Gemini	26
Gambar 4.17 Virus Gemini	26
Gambar 4.18 Diagram Alir Sistem	28
Gambar 4.19 Diagram Alir Proses MKNN	29
Gambar 4.20 Diagram Alir Perhitungan Jarak Euclidean Antar Data Latih.....	30
Gambar 4.21 Diagram Alir Perhitungan Validitas Data Latih.....	31
Gambar 4.22 Diagram Alir Perhitungan Jarak Euclidean Data Latih dan Data Uji	32
Gambar 4.23 Diagram Alir Perhitungan <i>Weight Voting</i>	33
Gambar 4.24 Diagram Alir Penentuan Kelas Data Uji.....	34

Gambar 4.25 Desain Tampilan Awal	40
Gambar 4.26 Desain Tampilan Data Uji	40
Gambar 4.27 Desain Proses MK-NN	41
Gambar 5.1 Bagan Implementasi.....	42
Gambar 5.2 Antarmuka Awal.....	49
Gambar 5.3 Antarmuka Data Uji.....	49
Gambar 5.4 Antarmuka Proses MK-NN Sebelum Dilakukan Klasifikasi.....	50
Gambar 5.5 Tampilan Proses MK-NN Setelah Dilakukan Klasifikasi.....	50
Gambar 6.1 Grafik Hasil Pengujian Pengaruh Nilai K	65



DAFTAR SOURCE CODE

Source Code 5.1 Perhitungan Jarak Euclidean Antar Data Latih	44
Source Code 5.2 Perhitungan Nilai Validitas.....	46
Source Code 5.3 Perhitungan Jarak Euclidean Data Latih dan Data Uji	46
Source Code 5.4 Perhitungan <i>Weight Voting</i>	47
Source Code 5.5 Proses Klasifikasi	48



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Cabai merah merupakan salah satu jenis sayuran yang cukup penting di Indonesia, baik sebagai komoditas yang dikonsumsi di dalam negeri maupun sebagai komoditas ekspor. Sebagai sayuran, cabai merah selain memiliki nilai gizi yang cukup tinggi, juga mempunyai nilai ekonomi tinggi. (Harpenas dan Dermawan, 2011).

Produktivitas cabe merah nasional masih sangat rendah yaitu 7,34 ton/ha sedangkan untuk Propinsi Aceh sendiri baru mencapai 5,54 ton/ha hingga tahun 2011 (BPS Dirjen Hortikultura, 2012).

Padahal potensi hasil sebenarnya mampu mencapai 12 ton/ha. Untuk memenuhi kebutuhan yang terus meningkat setiap tahunnya, maka peningkatan produksi cabai merah perlu dilakukan melalui intensifikasi. Kebutuhan cabai di Indonesia dari tahun ke tahun terus mengalami peningkatan. Namun begitu, hingga saat ini produksi cabai di Indonesia masih belum dapat memenuhi kebutuhan masyarakat secara luas. Hal ini disebabkan karena produksinya yang fluktuatif dengan produktivitas yang tergolong rendah. Rendahnya produktivitas cabai tersebut diduga disebabkan oleh berbagai faktor, antara lain mutu benih yang kurang baik, tingkat kesuburan tanah yang semakin menurun, penerapan teknik budidaya yang kurang baik, serta adanya permasalahan hama dan penyakit tanaman (Warisno dan Dahana, 2010).

Dilihat dari kandungan gizi dan manfaat yang dimiliki cabai, maka cabai juga penting dikonsumsi oleh manusia. Untuk itu pembudidayaan tanaman cabai harus diperhatikan agar produksi tanaman cabai meningkat dari tahun ketahun. Dalam satu periode tanam, cabai dapat dipanen beberapa kali bila musim dan perawatannya baik dapat dipanen 15-17 kali, namun umumnya sebanyak 10-12 kali. Perawatan tanaman cabai lebih rumit dibandingkan dengan perawatan tanaman hortikultura lainnya, sehingga biaya perawatannya lebih mahal, rendahnya produksi juga dapat membuat harga cabai meningkat (Sunarjono, 2001). Rendahnya produksi cabai salah satunya disebabkan oleh adanya serangan hama dan penyakit karena dapat menyebabkan kerugian baik kualitas maupun kuantitas cabai. Salah satu penyakit yang mempengaruhi produksi tanaman cabai di Indonesia adalah penyakit virus yang menyerang cabai yaitu virus kuning dan virus keriting (Semangun, 2008). Virus dapat mempunyai bermacam-macam pengaruh terhadap tumbuhan, karena virus mempunyai daya tular yang tinggi karena itu virus semakin diakui sebagai kendala utama terhadap perkembangan tanaman cabai.

Suatu metode klasifikasi dapat diimplementasikan ke sebuah perangkat lunak untuk mengetahui penyakit tanaman cabai. Klasifikasi merupakan sebuah metode untuk mengelompokan data atau objek ke dalam kelas yang sebelumnya sudah didefinisikan. Data yang diinputkan dalam klasifikasi berupa sekumpulan atribut-atribut atau fitur-fitur yang merupakan ciri-ciri dari suatu data atau objek. Atribut atau fitur tersebut bisa berupa data diskrit maupun kontinyu. Ada bermacam-macam algoritme klasifikasi yang dapat digunakan saat ini, salah satu contohnya adalah *Modified K-Nearest Neighbor* (MKNN). Algoritme MKNN berasal dari algoritma *K-Nearest Neighbor* (K-NN) yang telah dimodifikasi dan memiliki kinerja yang lebih baik daripada K-NN. Pada algoritme MKNN, seluruh data latih harus divalidasi satu kali terlebih dahulu, yang dihitung sesuai dengan tetangganya. Selain menghitung validitas, dilakukan juga *weight voting* menggunakan nilai validasi tersebut (Parvin, 2010).

Beberapa penelitian yang menggunakan algoritme MKNN salah satunya untuk melakukan klasifikasi penyakit demam. Penelitian ini dilakukan karena adanya kemiripan gejala-gejala penyakit yang menimbulkan kesulitan dalam mendapatkan diagnosis sementara, yang membuat kurang tepatnya penanganan awal yang didapatkan pasien. Ada 15 gejala penyakit dengan total 133 jumlah data pasien penderita penyakit demam berdarah, tifoid dan malaria. Pengujian terhadap perubahan nilai k menghasilkan rata-rata akurasi sebesar 88,55%. Pengujian terhadap pengaruh variasi jumlah data latih menghasilkan rata-rata akurasi sebesar 92,42%. Pengujian pengaruh komposisi data latih menghasilkan rata-rata akurasi sebesar 87,89%, sementara pengujian pengaruh komposisi data latih dan data uji terhadap akurasi menghasilkan rata-rata akurasi sebesar 96,35% (Wafiyah et al, 2017).

Penelitian lainnya yang menggunakan algoritme MKNN adalah mengenai penentuan status gizi balita. Pada Penelitian tersebut dilakukan klasifikasi status gizi balita dengan 4 kelas klasifikasi gizi yaitu gizi baik, gizi kurang, gizi buruk dan gizi lebih, serta 4 atribut yaitu jenis kelamin, umur, berat badan dan tinggi badan. Akurasi penelitian ini berada di kisaran 80% (Khotimah, 2015).

Penelitian selanjutnya mengenai diagnosis penyakit kucing menggunakan algoritme MKNN. Penelitian ini melakukan klasifikasi penyakit pada kucing dengan jumlah data total sebanyak 220, 18 gejala dan 8 kelas penyakit kucing. Pengujian pengaruh nilai k menghasilkan akurasi maksimal sebesar 99,90%, sementara pengujian berdasarkan sebaran data menghasilkan akurasi maksimal sebesar 99,86%, dan hasil pengujian berdasarkan jumlah data latih menghasilkan akurasi maksimal sebesar 99,66% (Putri, 2016).

Berdasarkan uraian permasalahan yang telah disebutkan dan pemaparan penelitian sebelumnya, maka judul yang diusulkan dalam penelitian ini adalah "Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)".

1.2 Rumusan Masalah

Rumusan masalah penelitian yang didapatkan berdasarkan uraian latar belakang adalah sebagai berikut:

1. Bagaimana implementasi metode *Modified K-Nearest Neighbor (MKNN)* untuk diagnosis penyakit pada tanaman cabai.
2. Bagaimana hasil pengujian metode *Modified K-Nearest Neighbor (MKNN)* untuk diagnosis penyakit pada tanaman cabai.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Menerapkan metode *Modified K-Nearest Neighbor (MKNN)* untuk diagnosis penyakit pada tanaman cabai.
2. Menguji tingkat akurasi metode *Modified K-Nearest Neighbor (MKNN)* untuk diagnosis penyakit pada tanaman cabai.

1.4 Manfaat

Penelitian ini diharapkan dapat membantu para Petani untuk melakukan deteksi dini penyakit tanaman cabai sehingga dapat dilakukan tindakan penanganan awal terhadap penyakit cabai secara cepat dan tepat.

1.5 Batasan Masalah

Berdasarkan dengan latar belakang masalah, maka diberikan batasan yang digunakan sebagai pedoman dalam pelaksanaan penelitian ini, yang diantaranya:

1. Area Penelitian dilakukan di Badan Pengkajian Teknologi Pertanian (BPTP) Karang Ploso.
2. Penelitian hanya dilakukan pada tanaman cabai merah.
3. Menggunakan metode *Modified K-Nearest Neighbor (MKNN)*.
4. Penelitian langsung dilakukan kepada pakar pertanian di BPTP Karang Ploso.

1.6 Sistematika Pembahasan

Sistematika penyusunan penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi uraian tentang Latar Belakang, Rumusan Masalah, Tujuan, Manfaat, Batasan Masalah dari diagnosis penyakit cabai menggunakan metode *Modified K-Nearest Neighbour (MKNN)*.

BAB II LANDASAN TEORI

Bab ini berisi uraian kajian pustaka dari penelitian-penelitian sebelumnya terkait implementasi metode *Modified K-Nearest Neighbour (MKNN)*. Bagian ini juga memuat dasar teori yang berisi teori-teori lainnya yang berhubungan dengan penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi bahasan metodologi yang digunakan dalam penelitian yang terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, serta pengambilan kesimpulan dan saran.

BAB IV PERANCANGAN

Bab ini berisi bahasan mengenai rancangan implementasi sistem klasifikasi penyakit tanaman cabai menggunakan metode *Modified K-Nearest Neighbour (MKNN)*.

BAB V IMPLEMENTASI

Pada Bab ini berisi implementasi dari hasil rancangan yang sudah dibuat sebelumnya serta penjelasan langkah-langkah melakukan implementasi *Modified K-Nearest Neighbour (MKNN)* untuk mengklasifikasi penyakit cabai.

BAB VI PENGUJIAN

Pada Bab ini berisi penjelasan langkah-langkah pengujian sistem serta analisis hasil pengujian *Modified K-Nearest Neighbour (MKNN)* dalam melakukan klasifikasi penyakit cabai.

BAB VII KESIMPULAN DAN SARAN

Pada Bab ini berisi kesimpulan yang didapat berdasarkan hasil penelitian dari pengujian menggunakan *Modified K-Nearest Neighbour (MKNN)* serta pemberian saran untuk pengembangan penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi penjelasan singkat penelitian-penelitian sebelumnya juga pemaparan dasar-dasar teori untuk menunjang penyusunan penelitian ini. Beberapa teori yang dibutuhkan meliputi kajian pustaka, *data mining*, klasifikasi, metode *Modified K-Nearest Neighbour* (MKNN), dan penyakit tanaman cabai.

2.1 Kajian Pustaka

Kajian pustaka membahas tentang penelitian-penelitian sebelumnya yang menggunakan metode *Modified K-Nearest Neighbour* (MKNN) dengan berbagai macam objek. Beberapa penelitian yang menggunakan algoritme MKNN salah satunya untuk melakukan klasifikasi penyakit demam. Penelitian ini dilakukan karena adanya kemiripan gejala-gejala penyakit yang menimbulkan kesulitan dalam mendapatkan diagnosis sementara, yang membuat kurang tepatnya penanganan awal yang didapatkan pasien. Ada 15 gejala penyakit dengan total 133 jumlah data pasien penderita penyakit demam berdarah, tifoid dan malaria. Pengujian terhadap perubahan nilai k menghasilkan rata-rata akurasi sebesar 88,55%. Pengujian terhadap pengaruh variasi jumlah data latih menghasilkan rata-rata akurasi sebesar 92,42%. Pengujian pengaruh komposisi data latih menghasilkan rata-rata akurasi sebesar 87,89%, sementara pengujian pengaruh komposisi data latih dan data uji terhadap akurasi menghasilkan rata-rata akurasi sebesar 96,35% (Wafiyah et al, 2017).

Penelitian lainnya yang menggunakan algoritme MKNN adalah mengenai penentuan status gizi balita. Pada Penelitian tersebut dilakukan klasifikasi status gizi balita dengan 4 kelas klasifikasi gizi yaitu gizi baik, gizi kurang, gizi buruk dan gizi lebih, serta 4 atribut yaitu jenis kelamin, umur, berat badan dan tinggi badan. Akurasi penelitian ini berada di kisaran 80% (Khotimah, 2015).

Penelitian selanjutnya mengenai diagnosis penyakit kucing menggunakan algoritme MKNN. Penelitian ini melakukan klasifikasi penyakit pada kucing dengan jumlah data total sebanyak 220, 18 gejala dan 8 kelas penyakit kucing. Pengujian pengaruh nilai k menghasilkan akurasi maksimal sebesar 99,90%, sementara pengujian berdasarkan sebaran data menghasilkan akurasi maksimal sebesar 99,86%, dan hasil pengujian berdasarkan jumlah data latih menghasilkan akurasi maksimal sebesar 99,66% (Putri, 2016).

2.2 Tanaman Cabai

Cabai (*Capsicum annum* L) merupakan salah satu komoditas sayuran yang banyak dibudidayakan oleh petani di Indonesia karena memiliki harga jual yang tinggi dan memiliki beberapa manfaat kesehatan yang salah satunya adalah zat capsaicin yang berfungsi dalam mengendalikan penyakit kanker. Selain itu kandungan vitamin C yang cukup tinggi pada cabai dapat memenuhi kebutuhan harian setiap orang, namun harus dikonsumsi secukupnya untuk menghindari nyeri lambung.

Cabai memiliki sejarah serangan hama dan penyakit yang cukup banyak. Sebut saja hama thrips, lalat buah, kutu daun, dan tungau yang banyak meresahkan petani. Tidak hanya itu, penyakit seperti layu bakteri, layu fusarium, antraknosa, dan penyakit virus kuning bahkan lazim ditemui di beberapa sentra produksi cabai. Serangan hama dan penyakit ini berpotensi menurunkan produksi cabai. Bahkan pada beberapa kasus, serangan hama dan penyakit mampu menyebabkan gagal panen. Terkadang petani mengetahui ketika tanamannya diserang hama atau penyakit, tetapi petani tidak tahu hama atau penyakit apa yang sedang menyerang tanamannya. Masalah ini berkaitan erat dalam upaya pengendalian hama dan penyakit tanaman cabai besar secara strategis. Seorang pakar dibutuhkan untuk bertindak sebagai media bantu, mengingat terbatasnya pengetahuan para praktisi petani cabai dan kurangnya tenaga penyuluh.

Petani cabai merah sangat intensif menggunakan pestisida dalam mengendalikan hama dan penyakit, yang berakibat pada timbulnya masalah pencemaran air dan lingkungan, keracunan pada manusia atau hewan piaraan, terbunuhnya musuh alami, timbulnya peledakan hama baru, dan meningkatnya biaya produksi, sehingga dapat memperlemah daya jual. Dalam kaitannya dengan pengendalian hama dan penyakit cabai merah, telah dikembangkan Pengelolaan Hama Terpadu (PHT) yang dasarnya adalah menanam tanaman sehat sesuai dengan agroekosistemnya, dimulai dari pemilihan bibit/ benih yang sehat, cara penanaman sampai cara pemeliharaan, sehingga populasi hama dan penyakit tetap di bawah ambang kendali (AK). Konsep PHT ini jika diterapkan dengan baik dan konsisten dapat mengurangi penggunaan pestisida, dan ikut mengamankan lingkungan.

Pada kasus tertentu, tanaman cabai yang terlihat dalam kondisi yang ideal dan terlihat sehat. Namun ternyata tidak, tiba-tiba dapat terkena penyakit dan tidak jarang mati karena terlambat dalam penanganannya. Beberapa penyakit utama pada tanaman cabai adalah :

1. Busuk buah atau Antraknosa (*Colletotrichum gloeosporioides*)

Cendawan penyebab penyakit ini berkembang dengan spora yang berbentuk oval dengan ujung tumpul atau bengkok seperti sabit. Buah sakit ditandai adanya bercak coklat kehitaman pada permukaannya, kemudian busuk lunak. Pada bagian tengah bercak terdapat kumpulan titik hitam yang merupakan kelompok aservulus dan spora. Serangan berat menyebabkan seluruh buah

keriput dan kering. Warna kulit buah seperti jerami padi. Cuaca yang panas dan basah mempercepat perkembangan penyakit

2. Bercak daun (*Cercospora capsici*)

Serangan cendawan menimbulkan bercak kecil yang berbentuk bulat dan kering. Bercak meluas sampai garis tengahnya 0,5 cm dengan pusat bercak berwarna pucat sampai putih, tepi bercak berwarna gelap. Bagian tengah bercak rapuh dan mudah rusak menyebabkan daun berlubang. Apabila bercaknya banyak, maka daun menjadi kuning dan gugur. Cendawan ini juga menyerang batang dan tangkai buah. Sporangya dapat terbawa biji dan bertahan pada sisa tanaman sakit selama musim buah. Cuaca yang panas dan basah membantu perkembangan penyakit. Kadang-kadang penyakit menyerang tanaman di pembibitan.

3. Layu Fusarium (*Fusarium oxysporum*)

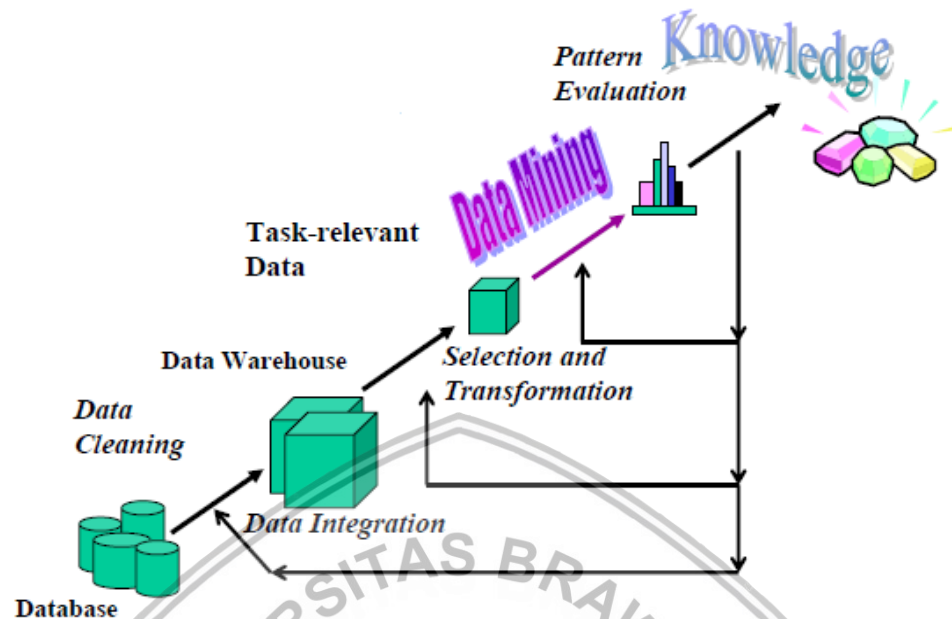
Tanaman yang terserang menjadi layu dimulai dari daun bagian bawah. Anak tulang daun tampak kuning, tanaman menjadi layu dengan cepat pada waktu 2 – 3 hari. Jaringan akar dan pangkal batang berwarna coklat. Tempat luka infeksi ditutupi oleh hifa yang berwarna putih seperti kapas.

4. Penyakit virus (Virus mosaik ketimun= CMV; virus mosaik tembakau = TMV; virus kentang Y= PVY)

Virus menyebabkan warna daun menjadi mosaik atau belang, ukuran lebih kecil dari daun normal. Bila menyerang tanaman muda, tanaman tumbuh kerdil.

2.3 Data Mining

Data mining, atau biasa disebut sebagai *Knowledge Discovery in Databases* (KDD), mengacu pada ekstraksi, penggalian, atau penemuan suatu pengetahuan (*knowledge*) yang berasal dari data yang berjumlah besar. Adapun beberapa yang menganggap bahwa data mining itu sendiri hanya merupakan suatu tahap penting dalam proses penemuan pengetahuan (Han, 2012). Gambar 2.1 menunjukkan proses penemuan pengetahuan.



Gambar 2.1 Proses Penemuan Pengetahuan

Sumber: Zaiane, 1999

Proses penemuan pengetahuan terdiri dari urutan iterasi langkah-langkah berikut.

- *Data cleaning*, untuk menghilangkan noise atau data yang tidak relevan dan tidak konsisten.
- *Data integration*, menggabungkan berbagai sumber data.
- *Data selection*, mengambil data dari basis data yang relevan dengan analisis.
- *Data transformation*, mengubah data ke dalam bentuk lain yang sesuai untuk ekstraksi.
- *Data mining*, proses penting yang menggunakan sebuah metode untuk ekstraksi pola data.
- *Pattern evaluation*, untuk mengidentifikasi pola menarik yang mewakili pengetahuan berdasarkan beberapa pengukuran.
- *Knowledge presentation*, menyajikan pengetahuan yang sudah diekstraksi kepada pengguna.

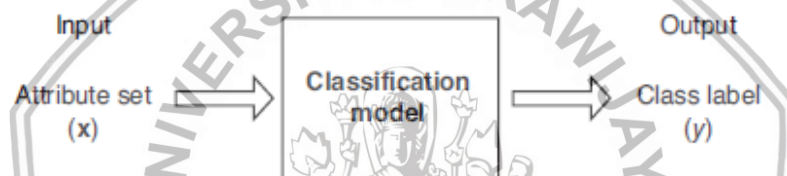
Berdasarkan langkah-langkah tersebut, *data mining* merupakan suatu langkah dalam proses penemuan pengetahuan yang sangat penting karena dapat menemukan suatu pola menarik. Oleh karena itu, dapat disimpulkan bahwa *data mining* adalah proses menemukan pengetahuan dan pola menarik dari sejumlah data yang berukuran besar. Data-data tersebut bisa bersumber dari basis data, *data warehouse*, web, dan penyimpanan informasi lainnya. Bentuk data yang umum digunakan untuk aplikasi *data mining* adalah basis data, *data warehouse*, dan data transaksional (Han, 2012).

Data mining memiliki beberapa fungsionalitas yang digunakan untuk menentukan jenis pola yang ditemukan dalam tugas *data mining*. Tugas tersebut

secara umum dikategorikan menjadi 2 macam yaitu deskriptif dan prediktif. Tugas deskriptif adalah untuk mengkarakterisasi properti-properti data dalam data set yang menjadi target, sedangkan prediktif adalah membuat prediksi dengan melakukan induksi pada data yang ada (Han, 2012).

2.4 Klasifikasi

Salah satu fungsionalitas *data mining* adalah klasifikasi. Klasifikasi merupakan proses yang terdiri dari dua langkah yaitu pembelajaran atau pelatihan (*learning*) dan klasifikasi (*classification*). Pembelajaran merupakan langkah membuat model klasifikasi sedangkan klasifikasi merupakan langkah menentukan kelas dari data yang diberikan menggunakan model yang sudah dibuat (Han, 2012). Proses klasifikasi ditunjukkan oleh Gambar 2.2



Gambar 2.2 Proses Klasifikasi

Sumber: Tan et al, 2006

Pada proses pelatihan, suatu set data latih dengan kelas yang telah diketahui dianalisis dan dibangun sebuah model dari setiap kelas dengan bantuan algoritme klasifikasi. Proses pelatihan pada klasifikasi disebut juga *supervised learning* karena setiap data latih sudah diketahui kelasnya masing-masing. Model yang didapatkan merupakan aturan-aturan klasifikasi (*classification rules*). Aturan-aturan ini kemudian diuji dengan data uji untuk memperkirakan akurasi. Data uji bersifat independen dan tidak digunakan dalam proses pelatihan. Bila akurasi baik, maka aturan-aturan tersebut dapat digunakan untuk klasifikasi data baru yang kelasnya tidak diketahui (Han, 2012).

2.5 Modified K-Nearest Neighbor (MKNN)

Algoritme *Modified K-Nearest Neighbor* (MK-NN) adalah pengembangan dari algoritme *K-Nearest Neighbor* (K-NN). K-NN sendiri merupakan salah satu metode sederhana yang sering digunakan dalam klasifikasi, di mana klasifikasi data uji dilakukan berdasarkan jarak k tetangga terdekatnya pada data latih. Sementara pada MK-NN, kelas label *query instance* ditentukan ke dalam k data latih yang sudah divalidasi dan kemudian dilakukan pembobotan pada data uji. Inilah yang menjadi perbedaan metode MK-NN dengan K-NN, di mana pada MK-NN terdapat proses perhitungan validitas seluruh data latih dan *weight voting*. Perhitungan validitas setiap data latih dilakukan satu kali berdasarkan setiap tetangganya. Nilai

validitas data latih ini nantinya digunakan sebagai informasi pada tahapan berikutnya. Berikut adalah *pseudo code* dari algoritme MK-NN (Parvin, 2010).

```

Output_label := MKNN (train_set, test_sample)
Begin
For i := 1 to train_size
Validity(i) := Compute validity of i-th sample;
End for;
Output_label := Weighted_KNN (Validity, test_sample);
Return Output_label;
End

```

Tahapan-tahapan proses klasifikasi menggunakan algoritme MK-NN adalah sebagai berikut.

1. Menentukan nilai k tetangga terdekat.
2. Menghitung jarak antar data latih dengan menggunakan persamaan jarak Euclidean yang ditunjukkan pada persamaan (2-1).

$$d_{(x,y)} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2-1)$$

Keterangan:

d = jarak
 i = variabel data
 x_i = sampel data
 y_i = data uji
 n = dimensi data

3. Menghitung validitas data latih.

Perhitungan validitas setiap data latih dilakukan berdasarkan pada setiap tetangga terdekatnya. Nilai validitas data latih ini digunakan untuk menghitung jumlah titik dengan label yang sama untuk data tersebut. Validitas data latih didefinisikan pada Persamaan (2-2).

$$Validitas(x) = \frac{1}{k} \sum_{i=1}^k S(label(x), label N_i(x)) \quad (2-2)$$

Keterangan:

k = jumlah titik terdekat
 $label(x)$ = kelas x
 $label N_i(x)$ = label kelas titik terdekat x

Nilai kesamaan antara titik x dan data ke- i dari tetangga terdekat ditentukan dengan menggunakan fungsi S pada persamaan (2-2). Fungsi S tersebut didefinisikan dengan persamaan (2-3).

$$S(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (2-3)$$

Keterangan:

a = kelas a pada data latih

b = kelas selain a pada data latih

a dan b adalah label kelas dari suatu data latih. Jika label kelas a dan b sama, maka S bernilai 1. Jika label kelas a dan b tidak sama, maka S bernilai 0.

4. Menghitung jarak antara data uji data latih menggunakan rumus jarak Euclidean seperti pada persamaan (2-1).
5. *Weight voting*

Salah satu variasi metode K-NN, *weighted K-NN*, menggunakan k tetangga terdekat, terlepas dari apapun kelasnya, namun kemudian menggunakan *weight voting* dari setiap sampel ketimbang *voting* mayoritas. Setiap sampel k diberikan bobot *voting* yang setara dengan beberapa fungsi menurun dari jaraknya yang berasal dari sampel yang tidak diketahui. Bobot setiap tetangga dihitung dengan persamaan (2-4).

$$W_{(i)} = \frac{1}{d + \alpha} \quad (2-4)$$

Keterangan:

$W_{(i)}$ = bobot setiap tetangga

d = jarak Euclidean data uji dengan data latih

α = *smoothing regulator*, bernilai 0,5

Voting pembobotan ini kemudian dijumlahkan untuk setiap kelas di mana kelas dengan total *vote* tertinggi dipilih. Kemudian validitas data latih dikalikan dengan bobot tersebut berdasarkan pada jarak Euclidean. Dari sini kemudian didapatkan persamaan (2-5) yaitu persamaan untuk melakukan *weight voting* pada MK-NN.

$$W_{(i)} = Validitas(i) \times \frac{1}{d + 0.5} \quad (2-5)$$

Keterangan:

$W_{(i)}$ = *weight voting*

$Validitas(i)$ = nilai validitas

d = jarak data uji dengan data latih

Weight voting memberikan pengaruh besar pada sampel yang mempunyai nilai validitas lebih tinggi dan lebih dekat dengan sampel uji. Mengalikan validitas dengan perhitungan bobot seperti pada persamaan (2-5) mampu mengatasi kelemahan dalam hal *outlier*. Karenanya, MK-NN merupakan metode yang lebih kuat dari metode K-NN biasa yang hanya berdasarkan pada jarak (Parvin, 2010).

6. Menentukan kelas dari data uji.

Kelas dari data uji ditentukan dengan mengambil nilai nilai *weight voting* terbesar sebanyak nilai k yang telah ditentukan. Dari nilai-nilai *weight voting* yang diambil, kelas data dari nilai yang paling besar merupakan kelas data uji.

2.6 Pengujian Akurasi

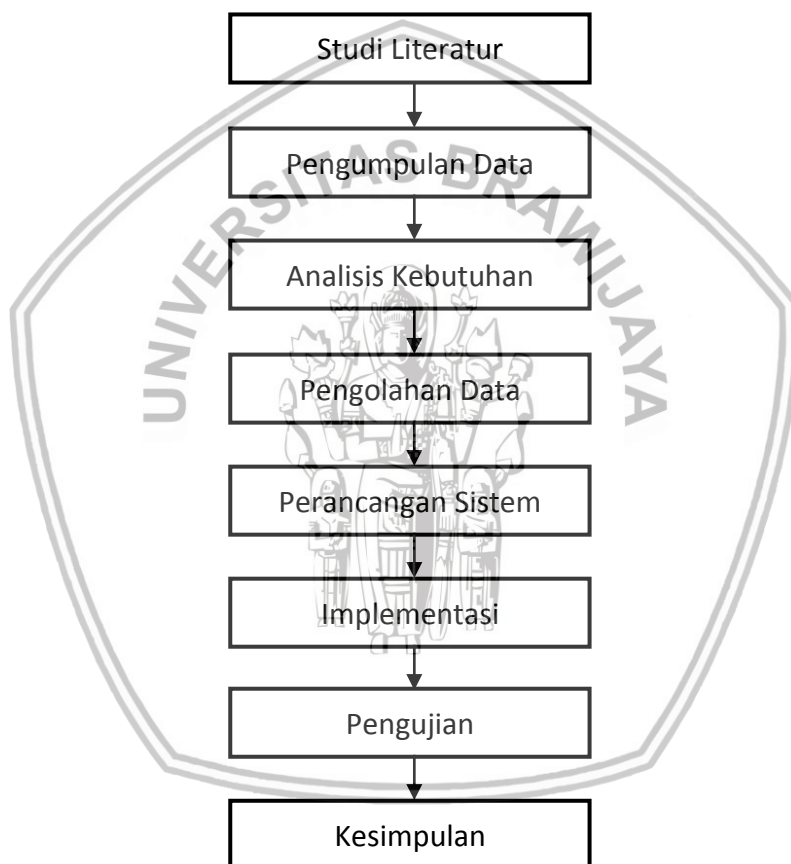
Pengujian akurasi adalah suatu ukuran kedekatan hasil pengukuran terhadap angka sebenarnya (*true value / reference value*). Pada penelitian ini pengujian akurasi dilakukan untuk mengetahui kemampuan sistem dalam membuat keputusan. Akurasi dilakukan dengan menghitung jumlah diagnosis yang tepat dibagi dengan jumlah data. Tingkat akurasi ini dapat diperoleh dengan perhitungan pada persamaan .

$$\text{Tingkat Akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100\% \quad (2-6)$$



BAB 3 METODOLOGI

Pada Bab ini akan dijelaskan langkah-langkah dalam penelitian “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)”. Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan yaitu studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian, analisis, pengambilan kesimpulan dan saran. Tahap pengerjaan ini diilustrasikan dengan blok metode penelitian seperti pada Gambar 3.1.



Gambar 3.1 Alur Metode Penelitian

3.1 Studi literatur

Studi literatur mempelajari literatur dari berbagai bidang ilmu yang menunjang penelitian tentang “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)” Diantaranya :

1. Sistem pakar.
2. Metode Modified K-NN.
3. Tanaman cabai.
4. Penyakit tanaman cabai.

3.2 Pengumpulan Data

Pada tahapan pengumpulan data, penelitian yang dibutuhkan adalah definisi penyakit cabai dan gejala-gejalanya. Sumber data diperoleh dari hasil wawancara yang dilakukan dengan seorang pakar dari BPTP Karang Ploso, penulis mendapatkan pengetahuan tentang gejala-gejala penyakit pada tanaman cabai.

Berdasarkan cara pengumpulan data untuk penelitian terdapat dua jenis data yaitu sekunder dan primer, data sekunder adalah data yang berasal dari orang lain dan tidak digunakan untuk kegiatan penelitian tetapi digunakan untuk tujuan penelitian seperti melalui buku literatur. Data primer merupakan sebuah data yang didapatkan langsung dari objek penelitian. Metode pengumpulan data yang didapatkan langsung dari objek penelitian. Metode pengumpulan data primer bersifat kuantitatif dapat menggunakan instrument kuisioner dan wawancara.

3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan proses identifikasi semua kebutuhan yang diperlukan dalam pengembangan “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)”. Analisis kebutuhan disesuaikan dengan variabel penelitian dan kebutuhan data yang akan digunakan.

Secara keseluruhan, kebutuhan yang digunakan dalam pengembangan sistem pada penelitian ini meliputi :

1. Data yang dibutuhkan, yaitu :
Data penyakit tanaman cabai.
2. Variabel yang digunakan untuk melakukan deteksi penyakit, yaitu :
Keadaan atau gejala yang terdapat pada tanaman cabai.

3.4 Pengolahan Data

Klasifikasi penyakit tanaman cabai membutuhkan data-data untuk dijadikan sebagai parameter pemrosesan data. Untuk mengklasifikasikan penyakit tanaman cabai terdapat 18 kriteria penilaian sebagai indikator pemeriksaan pada tanaman cabai.

Tabel 3.1 Kriteria Penilaian

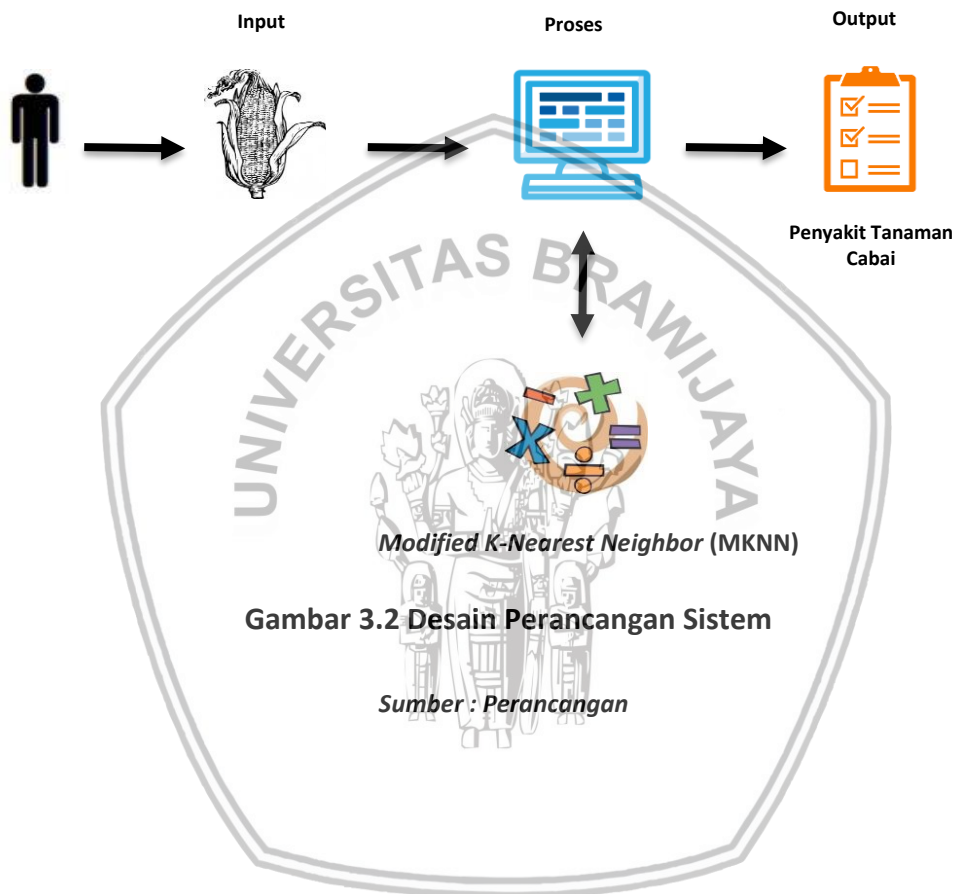
Kriteria	Penjelasan
G001	Bercak coklat kehitaman pada permukaan buah
G002	Bercak menjadi lunak
G003	Terdapat kumpulan titik-titik hitam
G004	Buah keriput dan mengering
G005	Warna kulit buah menjadi seperti jerami padi
G006	Bercak kecil yang berbentuk bulat dan kering
G007	Bercak meluas sampai garis tengahnya 0,5 cm dengan pusat bercak berwarna pucat putih, tepi bercak berwarna gelap
G008	Daun menguning kemudian meluas dan gugur
G009	Daun bagian bawah mulai layu
G010	Anak tulang daun menguning
G011	Tanaman menjadi layu
G012	Jaringan akar dan pangkal batang berwarna coklat
G013	Warna tulang daun berubah menjadi kuning terang
G014	Tulang daun menebal dan daun menggulung ke atas
G015	Daun mengecil dan berwarna kuning terang
G016	Produksi buah menurun dan lama kelamaan tidak berbuah
G017	Tanaman tumbuh kerdil
G018	Tanaman mati

3.5 Perancangan Sistem

Perancangan sistem dibangun berdasarkan hasil pengambilan data dari lapangan dan analisis kebutuhan yang telah dilakukan. Tahapan ini akan dijelaskan secara lengkap pada bab perancangan yang memuat tentang analisis kebutuhan perangkat lunak dan perancangan arsitektur sistem.

Pemodelan “Sistem Diagnosis Penyakit Tanaman Cabai Menggunakan Metode *Modified K-Nearest Neighbor* (MKNN)” pada penelitian kali ini menggunakan metode *Modified K-Nearest Neighbor* (MKNN) sebagai mesin inferensi. Keluaran sistem akan berupa jenis penyakit tanaman cabai berdasarkan perhitungan metode *Modified K-Nearest Neighbor* (MKNN) yang mempunyai nilai probabilitas terbesar.

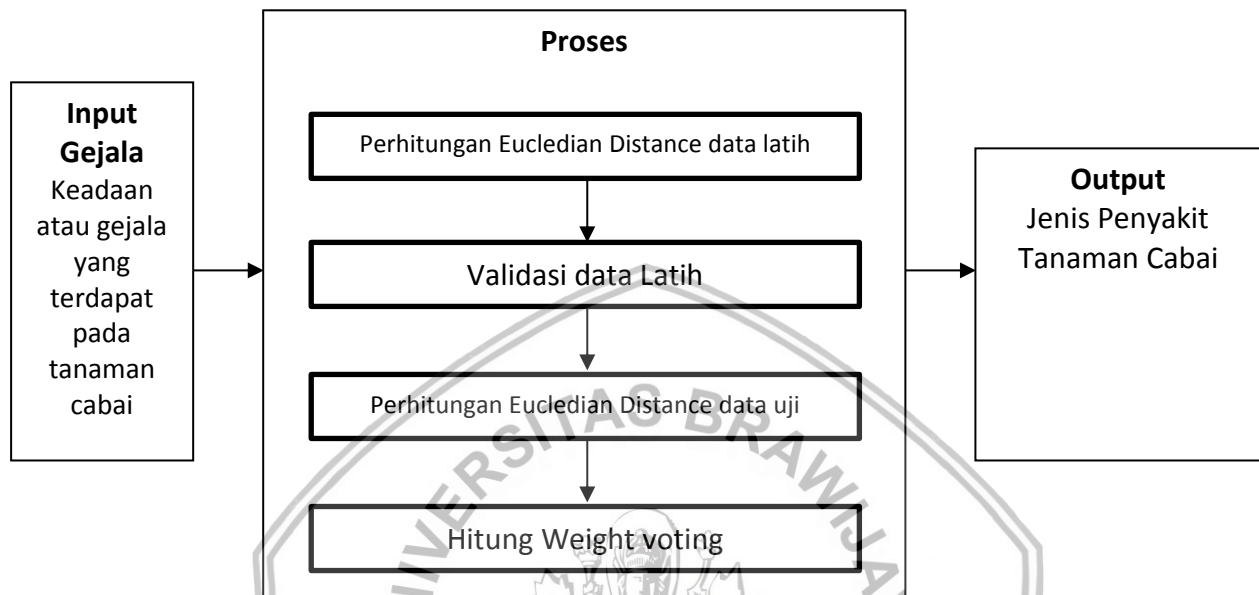
Gambar 3.2 menggambarkan Desain dari perancangan sistem. Pengembang adalah orang yang membuat sistem dan juga bertugas memasukkan nilai gejala dan penyakit sesuai data yang diberikan oleh pakar, Pengguna adalah aktor yang dapat mengakses sistem dimana pengguna nantinya akan memasukkan gejala ke sistem kemudian sistem akan memproses gejala yang telah dimasukkan oleh pengguna dengan metode *Modified K-Nearest Neighbor* (MKNN) kemudian menampilkan hasil diagnosis penyakit tanaman cabai.



Gambar 3.2 Desain Perancangan Sistem

Sumber : Perancangan

Gambar 3.3 merupakan sebuah diagram blok yang menguraikan fungsi-fungsi sistem dan menggambarkan cara kerja sistem secara keseluruhan. Adapun diagram blok dapat dilihat sebagai berikut:



Gambar 3.3 Blok Diagram Proses Sistem

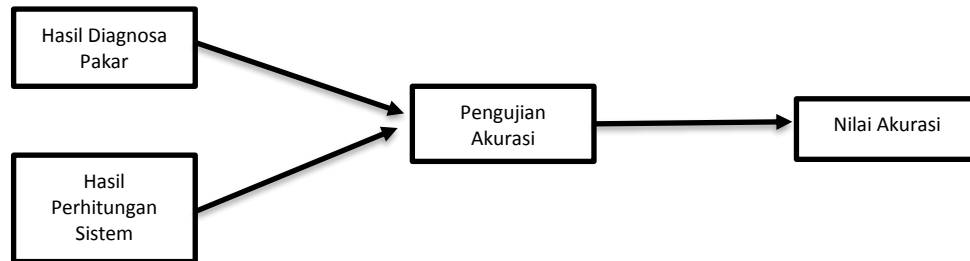
3.6 Implementasi

Implementasi sistem dilakukan dengan mengacu pada perancangan sistem yang telah ditentukan. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman JAVA, dan tools pendukung lainnya, implementasi sistem meliputi :

- Pembuatan *User Interface*.
- Penerapan metode *Modified K-Nearest Neighbor* (MKNN) dalam pengembangan program dengan bahasa pemrograman JAVA.

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui tingkat keberhasilan sistem yang telah dibangun. Pengujian sistem yang dilakukan adalah pengujian akurasi. Pengujian akurasi dilakukan dengan membandingkan hasil diagnosis sistem dengan hasil diagnosis seorang pakar. Alur pengujian akurasi ditunjukkan pada gambar 3.4 berikut :



Gambar 3.4 Diagram Blok Pengujian Akurasi

3.8 Analisis

Untuk analisis hasil dapat dilihat dari jawaban seorang pakar apakah sistem sudah berjalan dengan atau sistem sudah dapat mendeteksi penyakit secara benar dan tepat.

3.9 Kesimpulan

Setelah proses yang dimulai dari studi literatur sampai pengujian dan analisis hasil selesai dilakukan, hal terakhir yang perlu dilakukan adalah penarikan kesimpulan. Penarikan kesimpulan dilakukan dengan cara apakah sistem sudah dapat berjalan dengan benar dan apakah sudah dapat mendeteksi penyakit sesuai dengan diagnosis dari pakar. Selain penarikan kesimpulan dilakukan evaluasi terhadap penelitian ini untuk mengetahui kekurangan dan kesalahannya agar untuk kedepannya penelitian diagnosis penyakit tanaman cabai dapat dilakukan dengan lebih baik lagi dengan metode yang berbeda.

BAB 4 PERANCANGAN

Perancangan menjelaskan tentang kebutuhan yang diperlukan untuk membangun sistem diagnosis penyakit tanaman cabai menggunakan metode *Modified K-Nearest Neighbor* (MKNN).

4.1 Penyakit Tanaman Cabai

Cabai (*Capsicum annum* L) merupakan salah satu komoditas sayuran yang banyak dibudidayakan oleh petani di Indonesia karena memiliki harga jual yang tinggi dan memiliki beberapa manfaat kesehatan yang salah satunya adalah zat capsaicin yang berfungsi dalam mengendalikan penyakit kanker. Selain itu kandungan vitamin C yang cukup tinggi pada cabai dapat memenuhi kebutuhan harian setiap orang, namun harus dikonsumsi secukupnya untuk menghindari nyeri lambung.

Cabai memiliki sejarah serangan hama dan penyakit yang cukup banyak. Sebut saja hama thrips, lalat buah, kutu daun, dan tungau yang banyak meresahkan petani. Tidak hanya itu, penyakit seperti layu bakteri, layu fusarium, antraknosa, dan penyakit virus kuning bahkan lazim ditemui di beberapa sentra produksi cabai. Serangan hama dan penyakit ini berpotensi menurunkan produksi cabai. Bahkan pada beberapa kasus, serangan hama dan penyakit mampu menyebabkan gagal panen. Terkadang petani mengetahui ketika tanamannya diserang hama atau penyakit, tetapi petani tidak tahu hama atau penyakit apa yang sedang menyerang tanamannya. Masalah ini berkaitan erat dalam upaya pengendalian hama dan penyakit tanaman cabai besar secara strategis. Seorang pakar dibutuhkan untuk bertindak sebagai media bantu, mengingat terbatasnya pengetahuan para praktisi petani cabai dan kurangnya tenaga penyuluh.

Petani cabai merah sangat intensif menggunakan pestisida dalam mengendalikan hama dan penyakit, yang berakibat pada timbulnya masalah pencemaran air dan lingkungan, keracunan pada manusia atau hewan piaraan, terbunuhnya musuh alami, timbulnya peledakan hama baru, dan meningkatnya biaya produksi, sehingga dapat memperlemah daya jual. Dalam kaitannya dengan pengendalian hama dan penyakit cabai merah, telah dikembangkan Pengelolaan Hama Terpadu (PHT) yang dasarnya adalah menanam tanaman sehat sesuai dengan agroekosistemnya, dimulai dari pemilihan bibit/ benih yang sehat, cara pesemaian sampai cara pemeliharaan, sehingga populasi hama dan penyakit tetap di bawah ambang kendali (AK). Konsep PHT ini jika diterapkan dengan baik dan konsisten dapat mengurangi penggunaan pestisida, dan ikut mengamankan lingkungan.

Pada kasus tertentu, tanaman cabai yang terlihat dalam kondisi yang ideal dan terlihat sehat. Namun ternyata tidak, tiba-tiba dapat terkena penyakit dan tidak jarang mati karena terlambat dalam penanganannya. Beberapa penyakit utama pada tanaman cabai adalah :

1. Busuk buah atau Antraknosa (*Colletotrichum Gloeosporioides*)

Cendawan penyebab penyakit ini berkembang dengan spora yang berbentuk oval dengan ujung tumpul atau bengkok seperti sabit (Gambar 4.. Buah sakit ditandai adanya bercak coklat kehitaman pada permukaannya, kemudian busuk lunak (Gambar 4.2). Pada bagian tengah bercak terdapat kumpulan titik hitam yang merupakan kelompok aservulus dan spora. Serangan berat menyebabkan seluruh buah keriput dan kering. Warna kulit buah seperti jerami padi (Gambar 4.3 & Gambar 4.4). Cuaca yang panas dan basah mempercepat perkembangan penyakit.



Gambar 4.1 Busuk Buah



Gambar 4.2 Busuk Buah



Gambar 4.3 Busuk Buah



Gambar 4.4 Busuk Buah

2. Bercak daun (*Cercospora Capsici*)

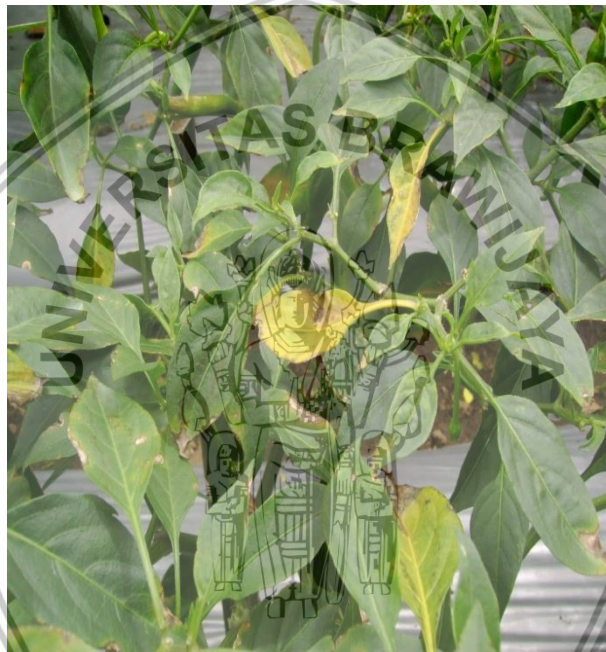
Serangan cendawan menimbulkan bercak kecil yang berbentuk bulat dan kering. Bercak meluas sampai garis tengahnya 0,5 cm dengan pusat bercak berwarna pucat sampai putih, tepi bercak berwarna gelap (Gambar 4.5). Bagian tengah bercak rapuh dan mudah rusak menyebabkan daun berlubang. Apabila bercaknya banyak, maka daun menjadi kuning dan gugur (Gambar 4.6). Cendawan ini juga menyerang batang dan tangkai buah. Sporanya dapat terbawa biji dan bertahan pada sisa tanaman sakit selama musim buah (Gambar 4.7). Cuaca yang panas dan basah membantu perkembangan penyakit. Kadang-kadang penyakit menyerang tanaman di pembibitan.



Gambar 4.5 Bercak Daun



Gambar 4.6 Bercak daun



Gambar 4.7 Bercak Daun

3. Layu Fusarium (*Fusarium Oxysporum*)

Tanaman yang terserang menjadi layu dimulai dari daun bagian bawah (Gambar 4.8). Anak tulang daun tampak kuning, tanaman menjadi layu dengan cepat pada waktu 2 – 3 hari (Gambar 4.11). Jaringan akar dan pangkal batang berwarna coklat (Gambar 4.9 & Gambar 4.10). Tempat luka infeksi ditutupi oleh hifa yang berwarna putih seperti kapas (Gambar 4.12 & Gambar 4.13).



Gambar 4.8 Layu Fusarium



Gambar 4.9 Layu Fusarium



Gambar 4.10 Layu Fusarium



Gambar 4.11 Layu Fusarium



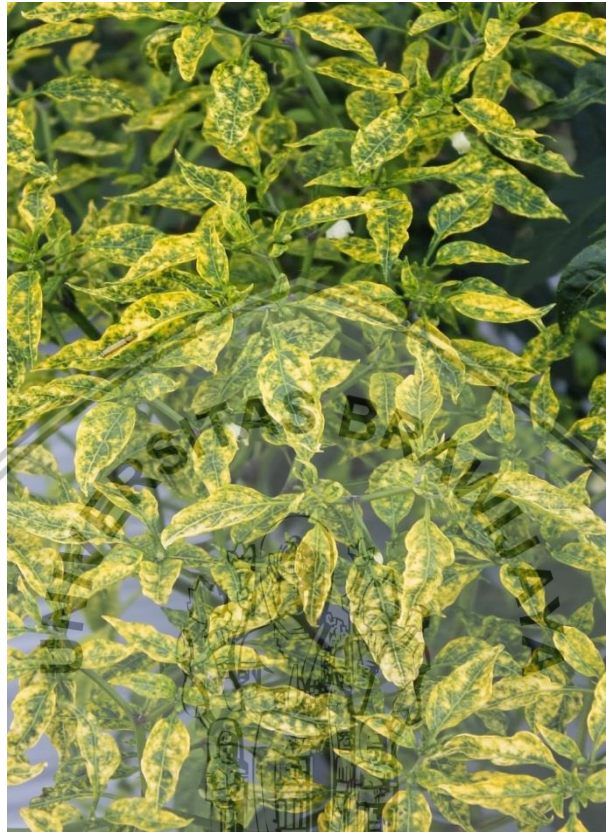
Gambar 4.12 Layu Fusarium



Gambar 4.13 Layu Fusarium

4. Penyakit Virus

Virus Gemini menyebabkan warna daun menjadi mosaik atau belang (Gambar 4.14). Ukuran lebih kecil dari daun normal (Gambar 4.16 & Gambar 4.17). Bila menyerang tanaman muda, tanaman tumbuh kerdil (Gambar 4.15).



Gambar 4.14 Virus Gemini



Gambar 4.15 Virus Gemini



Gambar 4.16 Virus Gemini



Gambar 4.17 Virus Gemini

Tabel penyakit Tanaman Cabai di tunjukkan pada tabel 4.1 dan 4.2

Tabel 4.1 Kode dan Gejala Penyakit Tanaman Cabai

Kode	Nama Gejala
G001	Bercak coklat kehitaman pada permukaan buah
G002	Bercak menjadi lunak
G003	Terdapat kumpulan titik-titik hitam
G004	Buah keriput dan mengering
G005	Warna kulit buah menjadi seperti jerami padi
G006	Bercak kecil yang berbentuk bulat dan kering
G007	Bercak meluas sampai garis tengahnya 0,5 cm dengan pusat bercak berwarna pucat putih, tepi bercak berwarna gelap
G008	Daun menguning kemudian meluas dan gugur
G009	Daun bagian bawah mulai layu
G010	Anak tulang daun menguning
G011	Tanaman menjadi layu
G012	Jaringan akar dan pangkal batang berwarna coklat
G013	Warna tulang daun berubah menjadi kuning terang
G014	Tulang daun menebal dan daun menggulung ke atas
G015	Daun mengecil dan berwarna kuning terang
G016	Produksi buah menurun dan lama kelamaan tidak berbuah
G017	Tanaman tumbuh kerdil
G018	Tanaman mati

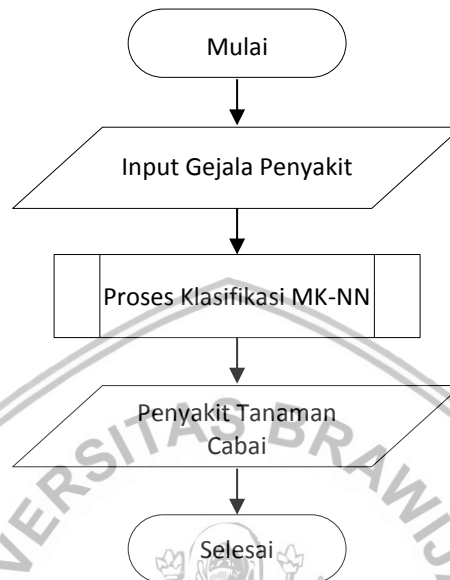
Tabel 4.2 Kode dan Penyakit Tanaman Cabai

Kode	Nama Penyakit
P1	Busuk Buah
P2	Bercak Daun
P3	Layu Fusarium
P4	Virus Gemini

4.2 Perancangan Proses

Untuk melakukan diagnosis penyakit tanaman cabai, diberikan masukan-masukan berupa gejala-gejala penyakit tanaman cabai dan nilai k pada sistem. Selanjutnya, dilakukan perhitungan jarak Euclidean antar data latih. Setelah jarak Euclidean didapatkan, dilakukan perhitungan validitas berdasarkan hasil jarak Euclidean. Selain melakukan perhitungan jarak Euclidean antar data latih, perhitungan jarak Euclidean antara data latih dengan data uji juga dilakukan. Hasil

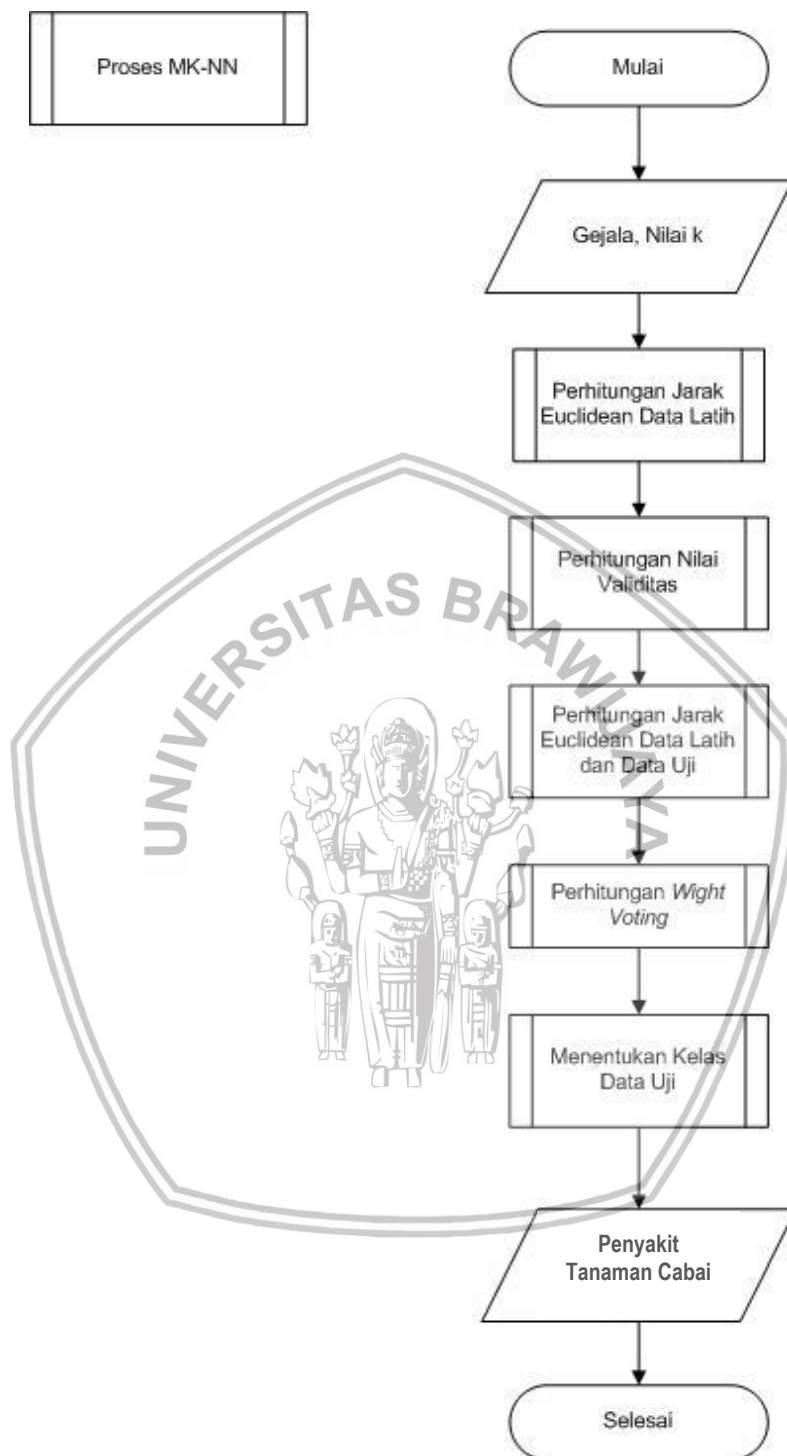
perhitungan yang didapatkan akan digunakan untuk perhitungan *weight voting*, yang selanjutnya digunakan untuk menentukan jenis penyakit dan masukan-masukan yang diberikan pada sistem. Secara umum, diagram alir sistem ditunjukkan pada Gambar 4.18.



Gambar 4.18 Diagram Alir Sistem

4.2.1 Proses Klasifikasi *Modified K-Nearest Neighbor* (MKNN)

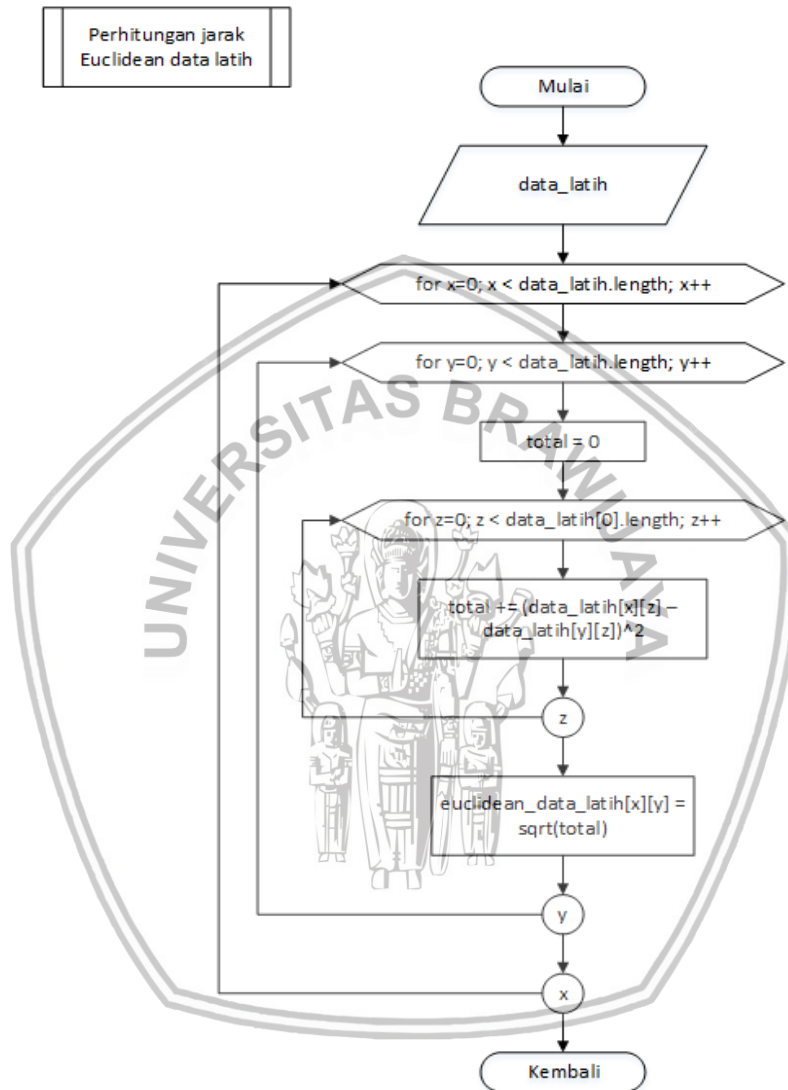
Dalam proses klasifikasi menggunakan *Modified K-Nearest Neighbor* (MKNN), terdapat proses perhitungan validitas, jarak Euclidean dan perhitungan *weight voting*. Proses klasifikasi dimulai dengan memasukan dataset gejala penyakit dan nilai k. Selanjutnya jarak Euclidean antar data latih dihitung. Hasil perhitungan jarak diurutkan dari yang terkecil kemudian diambil sejumlah k tetangga terdekat untuk perhitungan validitas. Setelah mendapatkan nilai validitas data latih, dilanjutkan dengan menghitung jarak Euclidean antara data latih dengan data uji. Hasil perhitungan jarak tersebut bersamaan dengan nilai validitas digunakan untuk perhitungan *weight voting*. Nilai *weight voting* terbesar yang didapatkan merupakan kelas penyakit. Diagram alir proses *Modified K-Nearest Neighbor* (MKNN) ditunjukkan pada Gambar 4.19.



Gambar 4.19 Diagram Alir Proses MKNN

4.2.2 Perhitungan Jarak Euclidean Antar Data Latih

Proses ini bertujuan untuk mendapatkan tetangga terdekat dari masing-masing data latih yang akan digunakan dalam perhitungan validitas. Tahapan-tahapan perhitungan jarak antar data latih ditunjukkan pada Gambar 4.20.

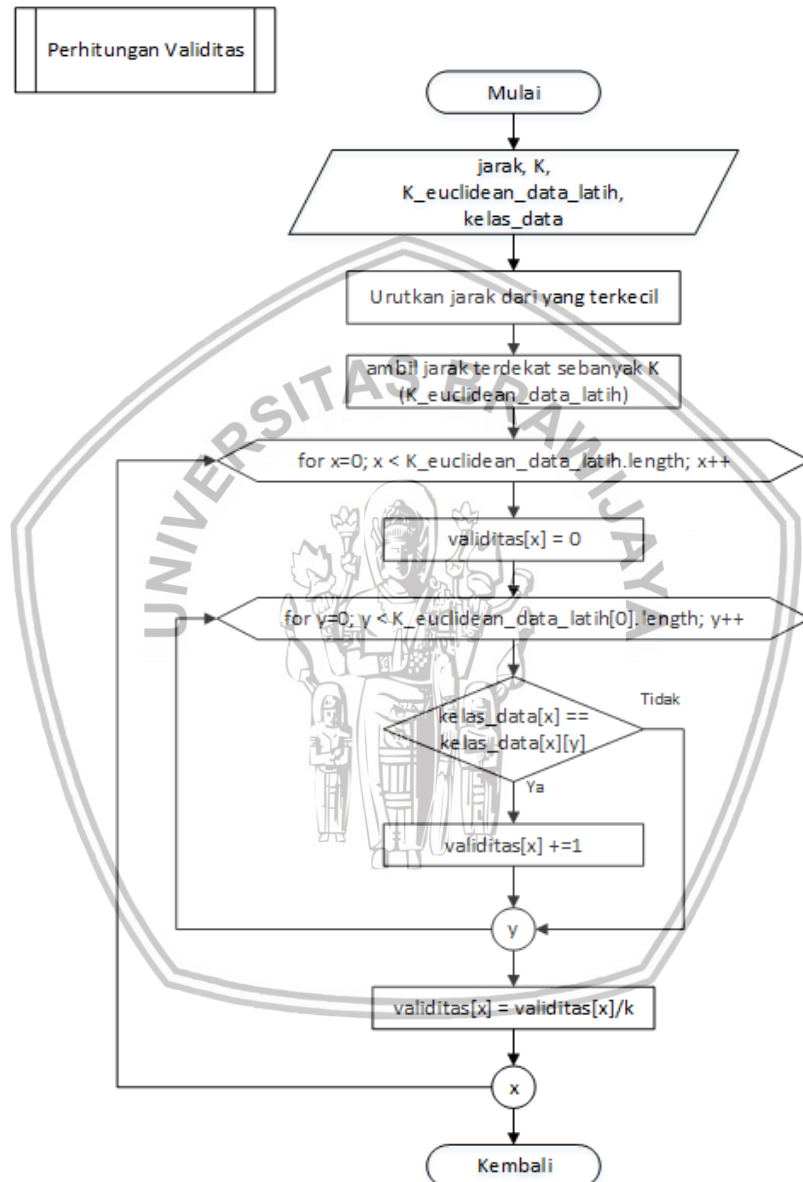


Gambar 4.20 Diagram Alir Perhitungan Jarak Euclidean Antar Data Latih

4.2.3 Perhitungan Validitas

Proses ini bertujuan mendapatkan nilai validitas data latih yang digunakan untuk perhitungan *weight voting*. Perhitungan validitas melibatkan jarak Euclidean antar data latih, nilai *k* dan kelas data latih. Jarak yang telah didapatkan kemudian diurutkan dari yang terkecil ke yang terbesar dan diambil yang memiliki nilai terkecil sejumlah nilai *k*, sehingga didapatkan tetangga terdekat dari setiap data latih. Selanjutnya, kelas data latih dengan kelas tetangga-tetangga

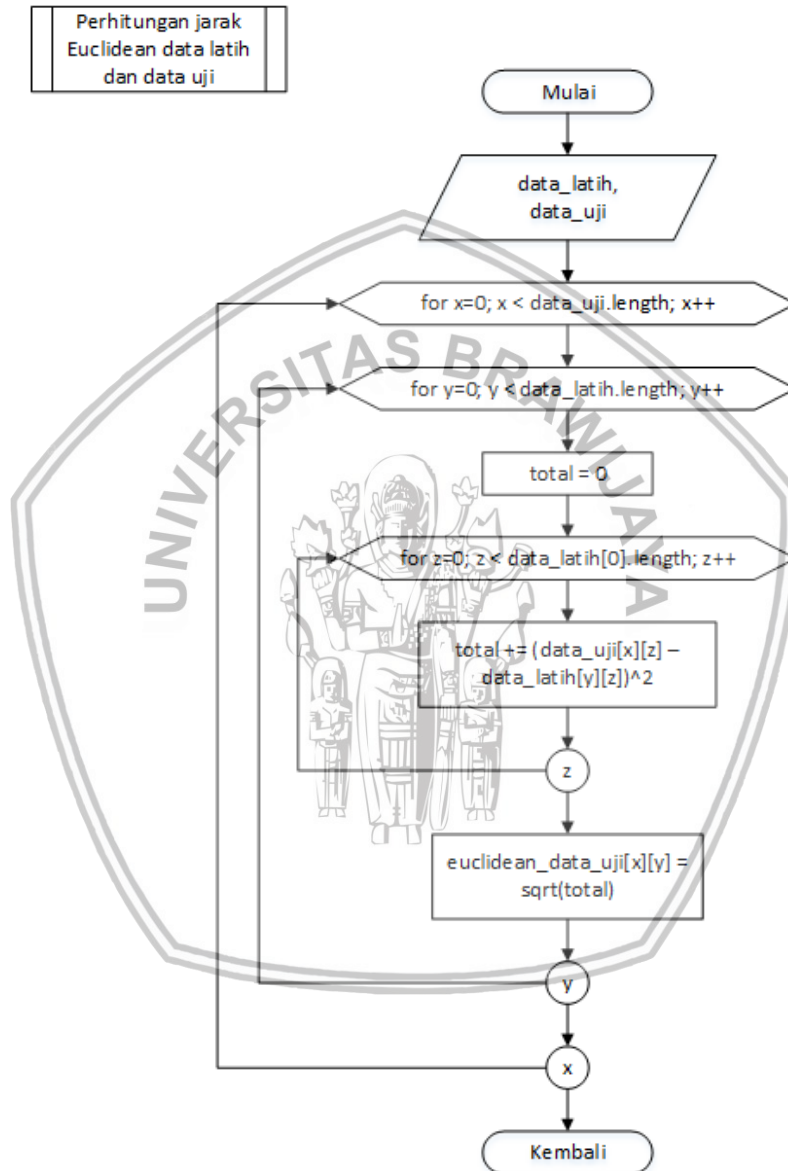
terdekatnya dibandingkan untuk mendapatkan nilai kesamaan. Jika kelasnya sama maka nilai kesamaannya 1, jika kelasnya berbeda maka nilai kesamaannya 0. Nilai kesamaan yang didapatkan dijumlahkan, lalu dibagi dengan nilai k. Hasil pembagian merupakan nilai validitas. Diagram alir perhitungan validitas ditunjukkan pada Gambar 4.21.



Gambar 4.21 Diagram Alir Perhitungan Validitas Data Latih

4.2.4 Perhitungan Jarak Euclidean Antara Data Latih dan Data Uji

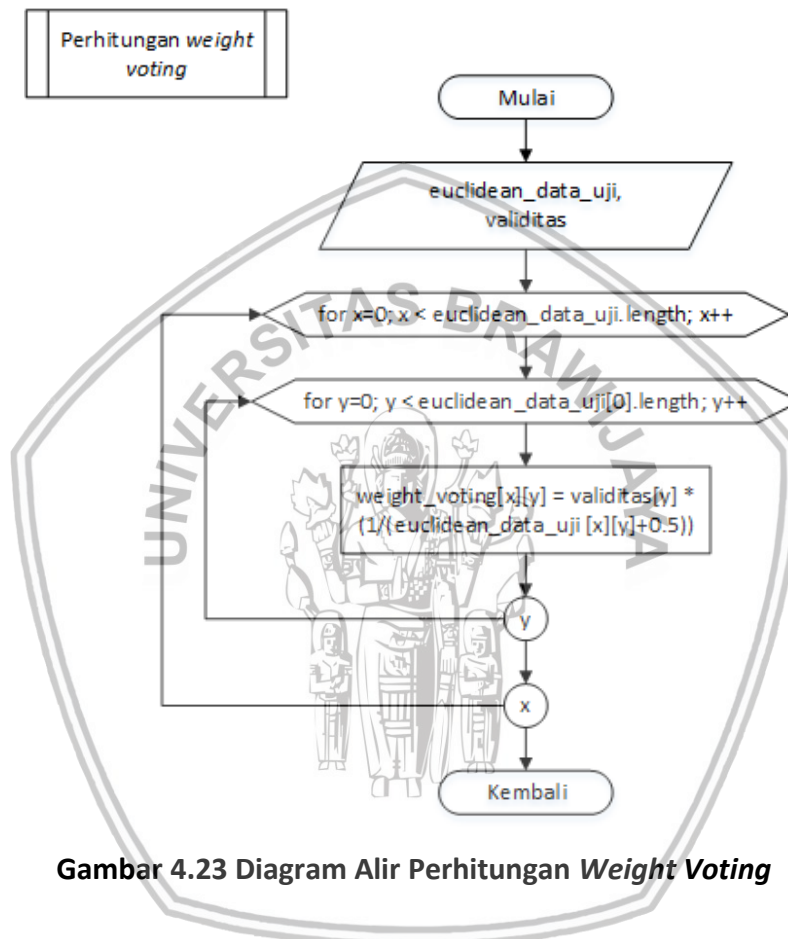
Proses ini bertujuan untuk mendapatkan jarak antara data latih dan data uji yang akan digunakan dalam perhitungan *weight voting* bersamaan dengan validitas. Diagram alir perhitungan jarak Euclidean antara data latih dan data uji ditunjukkan pada Gambar 4.22.



Gambar 4.22 Diagram Alir Perhitungan Jarak Euclidean Data Latih dan Data Uji

4.2.5 Perhitungan *Weight Voting*

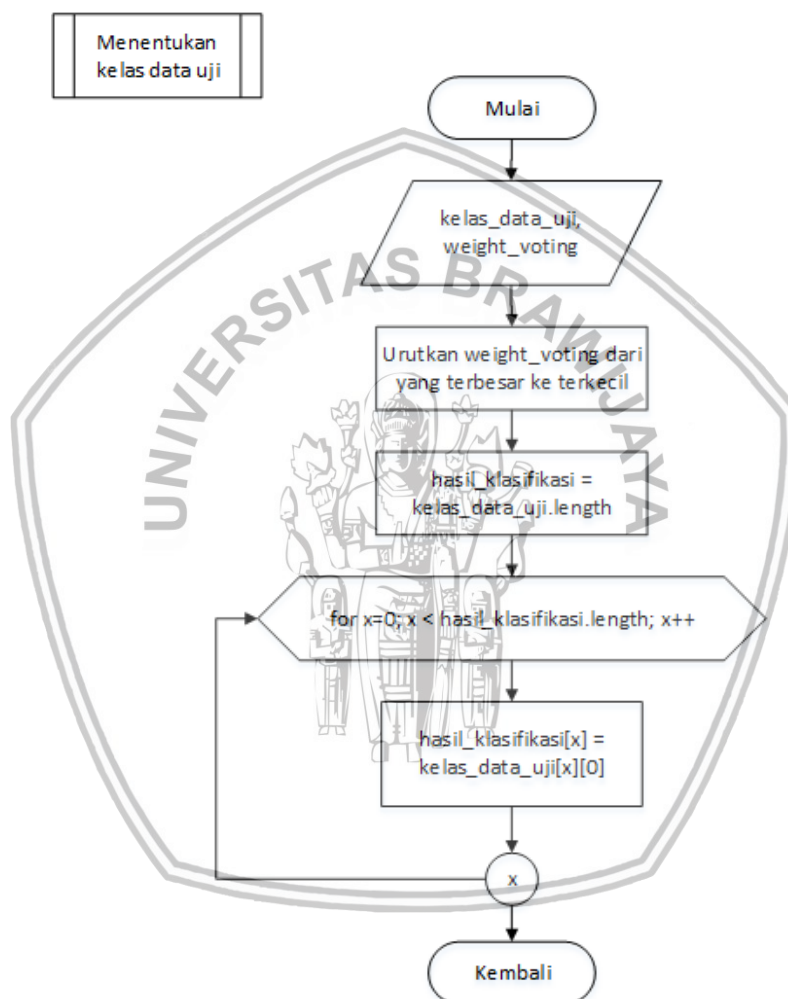
Perhitungan *weight voting* dilakukan menggunakan validitas data latih, jarak Euclidean antara data latih dengan data uji dan *smoothing regulator* yang bernilai 0.5. Nilai *weight voting* digunakan untuk menentukan kelas data uji. Diagram alir perhitungan *weight voting* ditunjukkan pada Gambar 4.23.



Gambar 4.23 Diagram Alir Perhitungan *Weight Voting*

4.2.6 Penentuan Kelas Data Uji

Setelah melakukan perhitungan *weight voting*, selanjutnya adalah menentukan kelas data uji. Hasil *weight voting* yang didapatkan diurutkan terlebih dahulu dari yang terbesar ke terkecil. Kelas dari nilai *weight voting* terbesar merupakan kelas dari data uji. Diagram alir penentuan kelas data uji ditunjukkan pada Gambar 4.24.



Gambar 4.24 Diagram Alir Penentuan Kelas Data Uji

4.3 Perhitungan Manual

Diagnosis penyakit tanaman cabai menggunakan *Modified K-Nearest Neighbor* (MK-NN) dilakukan mengikuti tahapan-tahapan berikut.

1. Penentuan nilai k tetangga terdekat.
2. Menghitung jarak Euclidean antar data latih.
3. Menghitung validasi data latih.
4. Menghitung jarak Euclidean antar data latih dan data uji.
5. Menghitung *weight voting*.
6. Menentukan kelas data uji

Dalam contoh perhitungan manual, diberikan 10 data latih dan 1 data uji, untuk data latih itu sendiri terdiri dari 5 kelas yaitu P1 hingga P5. Data latih dan data uji perhitungan manual ditunjukkan pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Data Latih

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	0	0	3	4	8	7	2	2	0	0
P1	0	0	2	3	8	8	3	4	0	0
P2	3	4	0	6	7	0	9	4	3	0
P2	0	1	1	5	6	9	6	4	4	3
P3	0	2	0	5	7	8	0	2	3	3
P3	1	0	9	3	0	4	1	2	0	3
P4	9	8	3	4	3	3	5	3	2	3
P4	9	9	1	1	2	3	5	4	3	3
P5	1	1	2	3	1	2	3	6	7	9
P5	2	2	4	4	5	4	3	6	9	9

Tabel 4.4 Data Uji

G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
0	0	0	4	3	7	8	5	3	0

Langkah 1. Menentukan nilai k

Nilai k yang digunakan dalam contoh perhitungan manual ini adalah 3.

Langkah 2. Menghitung jarak Euclidean antar data latih

Digunakan persamaan (2-1) untuk menghitung jarak Euclidean antar data latih. Hasilnya dapat dilihat pada Tabel 4.5.

Tabel 4.5 Jarak Euclidian Antar Data Latih

	1	2	3	4	5	6	7	8	9	10
1	0	2,8284	12,247	7,6811	6,1644	11	14,457	16,062	15	14,353
2	2,8284	0	12,166	6,7082	6,6332	12,124	14,663	15,811	14,866	14,283
3	12,247	12,166	0	11	13,115	16,155	10,817	12,083	14,248	14,213
4	7,6811	6,7082	11	0	6,7082	13,266	13,638	14,663	11,747	10,724
5	6,1644	6,6332	13,115	6,7082	0	12,845	13,964	15,033	12,61	11,747
6	11	12,124	16,155	13,266	12,845	0	14	15,716	12,649	13,892
7	14,457	14,663	10,817	13,638	13,964	14	0	4,1231	13,928	13,748
8	16,062	15,811	12,083	14,663	15,033	15,716	4,1231	0	13,964	14,353
9	15	14,866	14,248	11,747	12,61	12,649	13,928	13,964	0	5,5678
10	14,353	14,283	14,213	10,724	11,747	13,892	13,748	14,353	5,5678	0

Langkah 3. Melakukan sorting euclidian distance antar data latih dan menghitung validitas data latih

Setelah jarak Euclidean didapatkan, diambil jarak dengan nilai terkecil dari setiap data latih sejumlah k tetangga terdekat yang telah ditentukan yaitu 3. Jika kelas datanya sama maka nilainya 1, sedangkan jika kelas datanya berbeda maka nilainya 0. Euclidian antar data latih yang telah di urutkan berdasarkan nilai terkecil hingga terbesar ditunjukkan oleh tabel 4.6.

Tabel 4.6 Euclidian antar data latih yang telah diurutkan

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	2,8284	2,8284	10,817	6,7082	6,1644	11	4,1231	4,1231	5,5678	5,5678
3	6,1644	6,6332	11	6,7082	6,6332	12,124	10,817	12,083	11,747	10,724
4	7,6811	6,7082	12,083	7,6811	6,7082	12,649	13,638	13,964	12,61	11,747
5	11	12,124	12,166	10,724	11,747	12,845	13,748	14,353	12,649	13,748
6	12,247	12,166	12,247	11	12,61	13,266	13,928	14,663	13,928	13,892
7	14,353	14,283	13,115	11,747	12,845	13,892	13,964	15,033	13,964	14,213
8	14,457	14,663	14,213	13,266	13,115	14	14	15,716	14,248	14,283
9	15	14,866	14,248	13,638	13,964	15,716	14,457	15,811	14,866	14,353
10	16,062	15,811	16,155	14,663	15,033	16,155	14,663	16,062	15	14,353

Euclidian data latih pada baris pertama keseluruhan nilainya 0 karena itu adalah euclidian data latih dengan dirinya sendiri oleh karena nilai pada baris pertama tidak dianggap nilai yang diambil sebanyak K ditunjukkan oleh tabel 4.7.

Tabel 4.7 Euclidian antar data latih sebanyak K

	1	2	3	4	5	6	7	8	9	10
1	2,8284	2,8284	10,817	6,7082	6,1644	11	4,1231	4,1231	5,5678	5,5678
2	6,1644	6,6332	11	6,7082	6,6332	12,124	10,817	12,083	11,747	10,724

3	7,6811	6,7082	12,083	7,6811	6,7082	12,649	13,638	13,964	12,61	11,747
---	--------	--------	--------	--------	--------	--------	--------	--------	-------	--------

Validitas data latih dihitung dengan menjumlahkan nilai kesamaan kelas data terlebih dahulu, kemudian hasilnya dibagi dengan nilai k tetangga terdekat yang telah ditentukan yaitu 3. Hasil perhitungan validitas data latih dapat dilihat pada Tabel 4.21.

Berikut ini adalah contoh perhitungan validitas data latih 1 sesuai dengan persamaan (2-2), dengan nilai kesamaan sesuai pada Tabel 4.8.

$$\begin{aligned}
 Validitas(x) &= \frac{1}{k} \sum_{i=1}^k S(label(x), labelN_i(x)) \\
 &= \frac{1}{3} \times (1 + 0 + 0) \\
 &= 0,333333
 \end{aligned}$$

Tabel 4.8 Validitas Data Latih

Data Latih	Nilai Kesamaan 1	Nilai Kesamaan 2	Nilai Kesamaan 3	Jumlah	Validitas
1	1	0	0	2	0,3333
2	1	0	0	2	0,3333
3	0	1	0	2	0,3333
4	0	0	0	2	0
5	0	0	0	2	0
6	0	0	0	2	0
7	1	0	0	3	0,3333
8	1	0	0	2	0,3333
9	1	0	0	3	0,3333
10	1	0	0	0	0,3333

Langkah 4. Menghitung jarak Euclidean antara data latih dengan data uji,

Jarak Euclidean antar data latih dengan data uji dihitung menggunakan persamaan (2-1). Keseluruhan hasil perhitungan jarak antara data latih dengan data uji dapat dilihat pada Tabel 4.9.

Tabel 4.9 Euclidian Data latih dan Data Uji

Data Latih ke -	Jarak
1	9,380832
2	8,124038
3	9,797959
4	5,567764
5	10,19804
6	13,30413

7	13,89244
8	14,42221
9	12,60952
10	13,41641

Langkah 5. Melakukan perhitungan *weight voting*

Perhitungan *weight voting* dilakukan menggunakan nilai validitas dan jarak Euclidean data latih dengan data uji seperti pada persamaan (2-5). Hasil perhitungan *weight voting* seluruh data latih ditunjukkan pada Tabel 4.8. Berikut ini adalah contoh perhitungan *weight voting* data uji 1 untuk kolom pertama, menggunakan nilai validitas data latih 1 seperti yang ada pada Tabel 4.8 dan menggunakan jarak Euclidean antara data latih 1 dengan data uji 1 seperti yang ada pada Tabel 4.9.

$$\begin{aligned}
 W_{(1,1)} &= Validitas_{(1)} \times \frac{1}{d_{(1,1)} + 0,5} \\
 &= 0,6667 \times \frac{1}{9,380832 + 0,5} \\
 &= 0,033735
 \end{aligned}$$

Tabel 4.10 Weighted Voting

Jarak Latih dan Uji	Validitas	Weighted
9,380832	0,333333	0,033735
8,124038	0,333333	0,038652
9,797959	0,333333	0,032369
5,567764	0	0
10,19804	0	0
13,30413	0	0
13,89244	0,333333	0,02316
14,42221	0,333333	0,022338
12,60952	0,333333	0,025427
13,41641	0,333333	0,023953

Langkah 6. Mengurutkan *weighted voting* dan menentukan kelas data uji

Untuk menentukan kelas data uji, diambil nilai *weight voting* terbesar, oleh karena itu hasil dari *weighted voting* akan diurutkan terlebih dahulu dari nilai

terbesar hingga nilai terkecil kemudian kelas dari nilai weighted voting terbesar lah yang akan menjadi hasil klasifikasi kelas data uji. Weighted voting yang telah diurutkan ditunjukkan oleh tabel 4.10 dan hasil klasifikasi ditunjukkan oleh tabel 4.11.

Tabel 4.11 Nilai Weighted Voting yang telah diurutkan

Weighted Voting
0,038652
0,033735
0,032369
0,025427
0,023953
0,02316
0,022338
0
0
0

Nilai K	Weighted Voting	Kelas
3	0,038652	P1

Dari contoh perhitungan manual diatas maka untuk contoh kasus menggunakan data latih dan data uji diatas dihasilkan kesimpulan bahwa dengan data uji tersebut dikategorikan dalam kelas P1.

4.4 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk memudahkan pembuatan antarmuka di mana antarmuka sendiri merupakan penghubung antara pengguna dengan sistem. Sistem diagnosis penyakit tanaman cabai yang akan dibangun merupakan sistem menggunakan bahasa pemrograman java. Sistem akan memberikan keluaran berupa hasil perhitungan-perhitungan proses *Modified K-Nearest Neighbor* (MKNN) dan hasil diagnosis penyakit tanaman cabai berdasarkan masukan yang diberikan yang berasal dari pengguna. Masukan-masukan tersebut berupa penyakit. Masukan dari pengguna kemudian dibaca oleh sistem dan dilakukan perhitungan menggunakan MK-NN.

4.4.1 Desain Tampilan Awal

Tampilan awal merupakan tampilan pertama Sistem diagnosis penyakit tanaman cabai saat pengguna menjalankan perangkat lunak. Pada halaman awal terdapat judul perangkat lunak, menu Data, menu Proses *Modified K-Nearest Neighbor* (MKNN), menu Data Latih dan menu Data Uji. Saat aplikasi dijalankan

akan langsung menampilkan menu Data Latih yang berfungsi untuk memasukan data latih. Desain antarmuka tampilan awal ditunjukkan pada Gambar 4.25.

Gambar 4.25 Desain Tampilan Awal

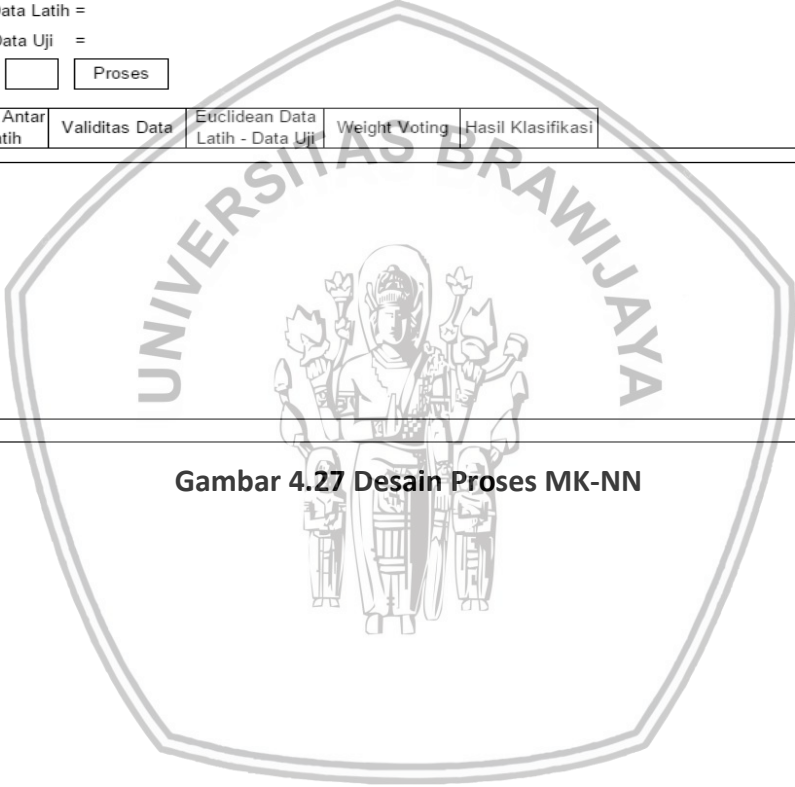
4.4.2 Desain Tampilan Data Uji

Bagian ini merupakan tampilan menu Data Uji yang berfungsi untuk memasukan data uji. Desain antarmuka tampilan data uji ditunjukkan pada Gambar 4.26.

Gambar 4.26 Desain Tampilan Data Uji

4.4.3 Desain Proses MK-NN

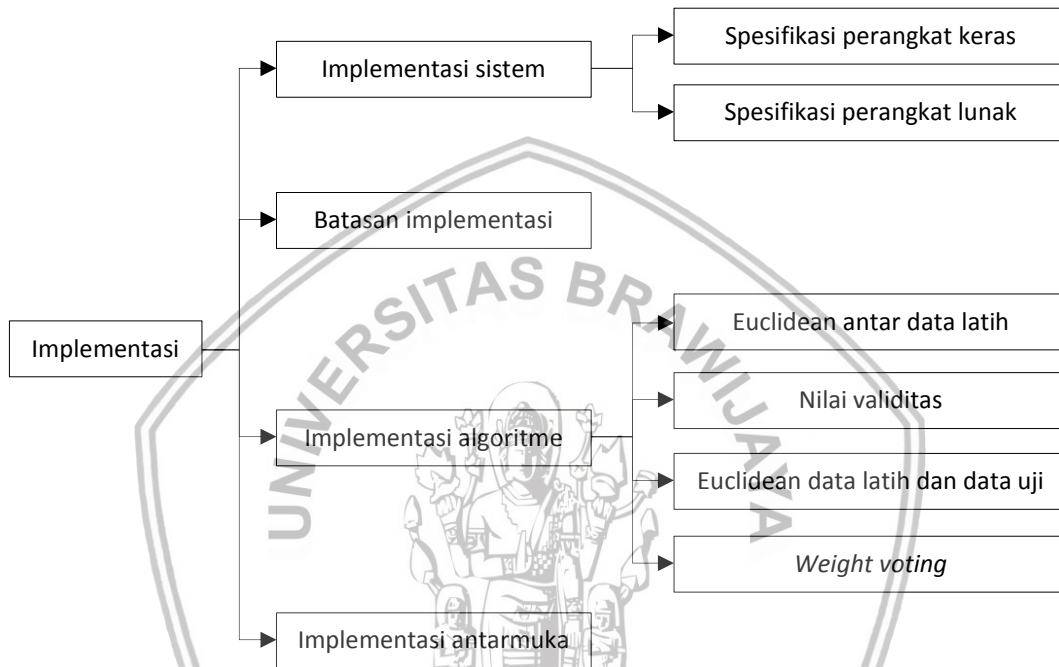
Bagian ini merupakan tampilan menu Proses *Modified K-Nearest Neighbor* (MKNN), di mana pengguna dapat memasukkan nilai k kemudian menekan *button* Proses untuk melakukan proses klasifikasi penyakit. Selain itu terdapat menu-menu lain yang berfungsi untuk menampilkan hasil perhitungan proses MK-NN, yaitu Euclidean Antar Data Latih, Validitas Data, Euclidean Data Latih – Data Uji, Weight Voting dan Hasil Klasifikasi, Desain Proses *Modified K-Nearest Neighbor* (MKNN) ditunjukkan pada Gambar 4.27.

Judul				
Data	Proses MK-NN			
Jumlah Data Latih = Jumlah Data Uji = Nilai K = <input type="text"/> <input type="button" value="Proses"/>				
Euclidean Antar Data Latih	Validitas Data	Euclidean Data Latih - Data Uji	Weight Voting	Hasil Klasifikasi
				

Gambar 4.27 Desain Proses MK-NN

BAB 5 IMPLEMENTASI

Berdasarkan analisis kebutuhan dan perancangan yang telah dilakukan pada sebelumnya, bagian ini akan membahas tentang implementasi algoritma *Modified K-Nearest Neighbor* (MKNN) untuk diagnosis penyakit tanaman cabai. Bahasan-bahasan dalam bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi algoritme, dan implementasi antarmuka, yang ditunjukkan pada Gambar 5.1.



Gambar 5.1 Bagan Implementasi

5.1 Implementasi Sistem

Berdasarkan Gambar 5.1, implementasi sistem memerlukan dua aspek penting agar sistem dapat berjalan dengan baik sesuai fungsinya. Dua aspek tersebut adalah perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem.

5.1.1 Spesifikasi Perangkat Keras

Tabel 5.1 menunjukkan daftar spesifikasi perangkat keras yang digunakan untuk membangun sistem.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i3-3320M CPU @ 2,60GHz
Memori	4 GB
Kartu grafis	Intel HD4000
Hardisk	320 GB

5.1.2 Spesifikasi Perangkat Lunak

Tabel 5.2 menunjukkan daftar spesifikasi perangkat lunak yang digunakan untuk membangun sistem.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem operasi	Windows 7
Bahasa pemrograman	Java
Tool pemrograman	Netbeans 8.1, JDK

5.2 Batasan Implementasi

Adapun batasan-batasan dalam implementasi algoritma *Modified K-Nearest Neighbor* (MKNN) untuk diagnosis penyakit tanaman cabai yaitu:

1. Sistem yang dibangun menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem disimpan dalam file txt.
3. Data yang digunakan berupa data-data gejala penyakit tanaman cabai, dan data penyakit tanaman cabai.
4. Keluaran dari sistem adalah salah satu dari 4 penyakit tanaman cabai.
5. Metode yang digunakan adalah *Modified K-Nearest Neighbor* (MKNN).

5.3 Implementasi Algoritma

Implementasi algoritme *Modified K-Nearest Neighbor* (MKNN) untuk diagnosis penyakit tanaman cabai terdiri dari perhitungan jarak Euclidean antar data latih, perhitungan nilai validitas, perhitungan jarak Euclidean antara data latih dengan data uji, dan *weight voting*.

5.3.1 Implementasi Perhitungan Jarak Euclidean Antar Data Latih

Langkah pertama dalam klasifikasi menggunakan *Modified K-Nearest Neighbor* (MKNN) adalah menghitung jarak antar data latih yang ditunjukkan pada Source Code 5.1.

```

1 public void Hitung_Euclidean_Data_Latih(int data_latih[][]){
2     this.euclidean_data_latih = new double [data_latih.length]
3     [data_latih.length-1];
4     for (int i=0; i < data_latih.length; i++){
5         int kolom = 0;
6         for (int j=0; j < data_latih.length; j++){
7             double total = 0.0;
8             if(i != j) {
9                 for (int k=0; k < data_latih[0].length; k++){

```

```

10         total+=(Math.pow(data_latih[i][k]-
11             data_latih[j][k],2));
12     }this.euclidean_data_latih[i][kolom]= Math.sqrt(total);
13     kolom++;
14 }}}}
15
16 public void Set_Kelas_Data_Latih(String kelas_dt_latih[]){
17     this.kelas_data_latih = new String[kelas_dt_latih.length]
18     [kelas_dt_latih.length-1];
19
20     for (int i=0; i < kelas_data_latih.length; i++){
21         int count = 0;
22         for (int j=0; j < kelas_dt_latih.length; j++){
23             if ( i != j){
24                 this.kelas_data_latih[i][count] = kelas_dt_latih[j];
25                 count++;
26             }
27         }
28     }

```

Source Code 5.1 Perhitungan Jarak Euclidean Antar Data Latih

Penjelasan Source Code 5.1 adalah sebagai berikut:

1. Baris 1 sampai 11 adalah kode untuk menghitung total jarak Euclidean antar data latih untuk setiap kolomnya.
2. Baris 12 adalah kode untuk mengakarkan total jarak Euclidean yang telah didapatkan sebelumnya.
3. Baris 16 sampai 27 adalah kode untuk set kelas dari data latih.

5.3.2 Implementasi Perhitungan Nilai Validitas

Sebelum melakukan perhitungan validitas, hasil perhitungan jarak Euclidean yang didapatkan diurutkan terlebih dahulu dari yang terkecil ke yang terbesar untuk mendapatkan tetangga terdekat. Setelah diurutkan, diambil jarak terkecil sejumlah nilai k yang dimasukan pengguna. Selanjutnya adalah membandingkan kelas setiap data latih dengan kelas dari tetangga terdekatnya. Jika kelasnya sama maka nilai kesamaannya 1 dan jika berbeda maka nilai kesamaannya 0. Hasilnya kemudian dijumlahkan dan dibagi dengan nilai k untuk mendapatkan nilai validitas. Implementasi algoritme perhitungan nilai validitas ditunjukkan pada Source Code 5.2.

```

1 public void Short_Euclidean_Data_Latih(){
2     double temp =0.0;
3     String temp_="";

```

```

4   for (int i=0; i < this.euclidean_data_latih.length; i++){
5       for (int j=0; j < this.euclidean_data_latih[0].length;
6           j++){
7           for (int k=0; k < (this.euclidean_data_latih[0].
8               length-1); k++){
9               if (this.euclidean_data_latih[i][k] >
10                  this.euclidean_data_latih[i][k+1]){
11                   temp = this.euclidean_data_latih[i][k];
12                   this.euclidean_data_latih[i][k] =
13                       this.euclidean_data_latih[i][k+1];
14                   this.euclidean_data_latih[i][k+1] = temp;
15                   temp_ = this.kelas_data_latih[i][k];
16                   this.kelas_data_latih[i][k] =
17                       this.kelas_data_latih[i][k+1];
18                   this.kelas_data_latih[i][k+1] = temp_;
19               }}}
20   }}
21
22   public void Ambil_Sebanyak_K(int K,
23   double euclidean_data_latih[][],
24   String kelas_data_latih[][]){
25       this.K_euclidean_data_latih =
26           new double[euclidean_data_latih.length][K];
27       this.K_kelas_data_latih =
28           new String[kelas_data_latih.length][K];
29       for (int i=0; i < K_euclidean_data_latih.length; i++){
30           for (int j=0; j < K_euclidean_data_latih[0].length;
31               j++){
32               this.K_euclidean_data_latih[i][j] =
33                   euclidean_data_latih[i][j];
34               this.K_kelas_data_latih[i][j] =
35                   kelas_data_latih[i][j];
36           }}}
37
38   public void Hitung_Validity(int K,
39   String K_kelas_data_latih[][], String Kelas_data[]){
40       this.validity = new double[K_euclidean_data_latih.length];
41       for (int i=0; i < K_euclidean_data_latih.length; i++){

```



```

42     This.validity[i] = 0.0;
43     for (int j=0; j < K_euclidean_data_latih[0].length; j++){
44         if (Kelas_data[i].equalsIgnoreCase
45             (K_kelas_data_latih[i][j])){
46             this.validity[i] +=1;
47         }}
48     this.validity[i] = this.validity[i]/K;
49 }}

```

Source Code 5.2 Perhitungan Nilai Validitas

Penjelasan Source Code 5.2 adalah sebagai berikut:

1. Baris 1 sampai 20 adalah kode untuk mengurutkan hasil perhitungan jarak Euclidean dari yang terkecil ke terbesar.
2. Baris 22 sampai 36 adalah kode untuk mengambil jarak Euclidean terkecil sejumlah nilai k yang diinputkan pengguna.
3. Baris 38 sampai 49 merupakan kode program untuk mendapatkan nilai validitas, di mana baris 44 sampai 47 merupakan kode program untuk memberikan nilai 1 apabila kelas data latih sama dengan kelas dari data dengan jarak Euclidean terkecil yang diambil. Sedangkan baris 48 merupakan kode untuk menghitung nilai validitas.

5.3.3 Implementasi Perhitungan Jarak Euclidean Data Latih dan Data Uji

Langkah berikutnya yaitu menghitung jarak Euclidean antara data latih dengan data uji yang ditunjukkan pada Source Code 5.3.

```

1  public void Hitung_Euclidean_DataLatih_DataUji(
2  int data_latih[][], int data_uji[][]){
3      this.euclidean_data_uji =
4          new double[data_uji.length][data_latih.length];
5
6      for (int i=0; i < data_uji.length; i++){
7          for (int j=0; j < data_latih.length; j++){
8              double tot = 0.0;
9              for (int k=0; k < data_latih[0].length; k++){
10                 tot += (Math.pow(data_uji[i][k]-data_latih[j][k],2));
11             } this.euclidean_data_uji[i][j] = Math.sqrt(tot);
12         }}

```

Source Code 5.3 Perhitungan Jarak Euclidean Data Latih dan Data Uji

Penjelasan Source Code 5.3 adalah sebagai berikut:

1. Baris 1 sampai 10 adalah kode untuk menghitung total jarak Euclidean antara data latih dengan data uji untuk setiap kolomnya.

- Baris 11 merupakan kode program untuk mengakarkan total jarak Euclidean yang telah didapatkan sebelumnya.

5.3.4 Implementasi Perhitungan *Weight Voting*

Langkah selanjutnya adalah perhitungan *weight voting* untuk mendapatkan kelas data uji. Perhitungan *weight voting* melibatkan nilai validitas dan jarak Euclidean data latih dengan data uji. Implementasi *weight voting* ditunjukkan pada Source Code 5.4.

```

1 public void Weighted_Voting(double Validitas[]){
2     for (int i=0; i < this.euclidean_data_uji.length; i++){
3         for (int j=0; j < this.euclidean_data_uji[0].length;
4             j++){
5             This.euclidean_data_uji[i][j] = Validitas[j]*
6                 (1./ (this.euclidean_data_uji[i][j]+0.5));
7         }}}
8
9 public void setKelas_Data_Uji(String kelas_dt_latih[],
10 int jumlah_data_uji){
11     this.kelas_data_uji = new String[jumlah_data_uji]
12         [kelas_data_latih.length];
13
14     for (int i=0; i < kelas_data_uji.length; i++){
15         for (int j=0; j < kelas_data_uji[0].length; j++){
16             this.kelas_data_uji[i][j] = kelas_dt_latih[j];
17         }}}

```

Source Code 5.4 Perhitungan *Weight Voting*

Penjelasan Source Code 5.4 adalah sebagai berikut:

- Baris 1 sampai 7 adalah kode untuk menghitung *weight voting* setiap data uji.
- Baris 9 sampai 17 adalah kode untuk set kelas perhitungan *weight voting* berdasarkan kelas dari data latih.

5.3.5 Proses Klasifikasi

Langkah terakhir adalah melakukan klasifikasi data uji. Pada proses ini, nilai *weight voting* diurutkan dari yang terbesar ke terkecil. Kelas dari *weight voting* terbesar merupakan kelas dari data uji. Implementasi proses klasifikasi ditunjukkan pada Source Code 5.5.

```

1 public void Short_Euclidean_DataLatih_Uji(){
2     double temp =0.0;
3     String temp_="";

```

```

4   for (int i=0; i < this.euclidean_data_uji.length; i++){
5       for (int j=0; j < this.euclidean_data_uji[0].length;
6           j++){
7           for (int k=0; k<(this.euclidean_data_uji[0].length-1);
8               k++){
9               if (this.euclidean_data_uji[i][k] <
10                  this.euclidean_data_uji[i][k+1]){
11                   temp = this.euclidean_data_uji[i][k];
12                   this.euclidean_data_uji[i][k] =
13                       this.euclidean_data_uji[i][k+1];
14                   this.euclidean_data_uji[i][k+1] = temp;
15
16                   temp_ = this.kelas_data_uji[i][k];
17                   this.kelas_data_uji[i][k] =
18                       this.kelas_data_uji[i][k+1];
19                   this.kelas_data_uji[i][k+1] = temp_;
20               }}}
21   }}
22
23   public void Hasil_Klasifikasi(String kelas_data_uji[][]){
24       this.hasil_klasifikasi =
25           new String[kelas_data_uji.length];
26       for (int i=0; i < this.hasil_klasifikasi.length; i++){
27           this.hasil_klasifikasi[i] = kelas_data_uji[i][0];
28   }}

```

Source Code 5.5 Proses Klasifikasi

Penjelasan Source Code 5.5 adalah sebagai berikut:

1. Baris 1 sampai 21 adalah kode untuk mengurutkan nilai *weight voting* dari yang terbesar ke yang terkecil.
2. Baris 23 sampai 28 adalah kode untuk set kelas data uji berdasarkan kelas dari *weight voting*.

5.4 Implementasi Antarmuka

Antarmuka bertujuan untuk memudahkan interaksi pengguna dengan sistem, Implementasi antarmuka terdiri dari tampilan antarmuka awal, data uji, proses *Modified K-Nearest Neighbor* (MKNN).

5.4.1 Implementasi Antarmuka Awal

Tampilan awal merupakan tampilan pertama saat pengguna menjalankan aplikasi. Pada halaman awal terdapat judul aplikasi, menu Data, menu Proses MKNN, menu Data Latih dan menu Data Uji. Saat aplikasi dijalankan akan langsung menampilkan menu Data Latih yang berfungsi untuk memasukan data latih. Pengguna dapat memasukan data latih berupa file .txt yang isinya akan ditampilkan pada aplikasi. Implementasi antarmuka awal ditunjukkan pada Gambar 5.2.

No.	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	KEL...
1	4	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
2	4	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
3	4	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
4	4	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
5	4	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
6	0	0	0	0	0	4	7	9	0	0	0	0	0	0	0	0	0	0	Berc...
7	0	0	0	0	0	4	7	0	0	0	0	0	0	0	0	0	0	0	Berc...
8	0	0	0	0	0	4	0	9	0	0	0	0	0	0	0	0	0	0	Berc...
9	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	Berc...
10	0	0	0	0	0	0	0	7	9	0	0	0	0	0	0	0	0	0	Berc...
11	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	4	Layu...
12	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	0	Layu...
13	0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	4	Layu...
14	0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	0	Layu...
15	0	0	0	0	0	0	0	0	5	6	0	9	0	0	0	0	0	4	Layu...
16	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	9	0	0	Viru...
17	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	0	0	Viru...
18	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	0	0	Viru...
19	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	0	0	Viru...
20	0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	8	9	0	Viru...

Gambar 5.2 Antarmuka Awal

5.4.2 Implementasi Antarmuka Data Uji

Menu Data Uji yang berfungsi untuk memasukan data uji. Pengguna dapat memasukan data uji berupa file .txt yang isinya akan ditampilkan pada aplikasi. Desain antarmuka tampilan data uji ditunjukkan pada Gambar 5.3.

No.	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	KEL...
1	4	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
2	4	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
3	4	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
4	4	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
5	4	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
6	4	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
7	4	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
8	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
9	4	0	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
10	4	0	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
11	4	0	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
12	4	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
13	4	0	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
14	4	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
15	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
16	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
17	0	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
18	0	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
19	0	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
20	0	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
21	0	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
22	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
23	0	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
24	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
25	0	0	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
26	0	0	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
27	0	0	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...
28	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bus...

Gambar 5.3 Antarmuka Data Uji

5.4.3 Implementasi Antarmuka Proses MK-NN

Menu Proses *Modified K-Nearest Neighbor* (MKNN) merupakan menu untuk klasifikasi data uji yang diberikan. Pada menu ini, pengguna dapat menginputkan nilai k kemudian menekan *button* Proses untuk melakukan klasifikasi. Pada menu ini, pengguna dapat melihat hasil perhitungan setiap proses MK-NN. Tampilan menu Proses MK-NN sebelum dilakukan klasifikasi dapat dilihat pada Gambar 5.4. Sedangkan tampilan menu Proses MK-NN setelah dilakukan klasifikasi dapat dilihat pada Gambar 5.5.

DIAGNOSIS PENYAKIT TANAMAN CABAI																				
Menggunakan Metode M-KNN																				
Data Proses MKNN																				
Jumlah Data Latih = 20																				
Jumlah Data Uji = 100																				
nilai K = 5 Proses																				
Euclidean Antar Data Latih Validitas Data Euclidean Data Uji-Data Latih Weighted Voting Hasil Klasifikasi																				
Euclidean Setelah Diurut					Kelas Data Setelah Euclidean Diurut							K-Euclidean Data					K-Kelas Data			
Euclidean Sebelum Terurut					Kelas Data Sebelum Euclidean Diurut															
No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
1	9	8	12.0...	7	19.7...	17.6...	18.52	16.1...	19.3...	21.6...	21.26	19.6...	19.2...	20.1	21.9...	20	20.4...	18.33	20.7...	
2	9	12.0...	8	11.4...	17.6...	15.1...	16.1...	13.4...	17.1...	19.6...	19.2...	17.4...	17.0...	17.9...	20	17.8...	18.33	15.9...	18.7...	
3	8	12.0...	9	10.63	18.1...	15.7...	16.7...	14.0...	17.5...	20.1	19.6...	17.9...	17.5...	18.4...	20.4...	18.33	18.7...	16.4...	19.1...	
4	12.0...	8	9	13.9...	15.7...	12.8...	14.0...	10.8...	15.1...	17.9...	17.5...	15.5...	15.0...	16.0...	18.33	15.9...	16.4...	13.82	16.9...	
5	7	11.4...	10.63	13.9...	18.52	16.1...	17.1...	14.5...	18.0...	20.4...	20.0...	18.3...	17.9...	18.8...	20.7...	18.7...	19.1...	16.9...	19.57	
6	19.7...	17.6...	18.1...	15.7...	18.52	9	7	11.4...	4	19.1...	18.7...	16.9...	16.4...	17.4...	19.5...	17.3...	17.8...	15.3...	18.2...	
7	17.6...	15.1...	15.7...	12.8...	16.1...	9	11.4...	7	9.849	16.9...	16.4...	14.3...	13.7...	14.9...	17.3...	14.7...	15.3...	12.45	15.8...	
8	18.52	16.1...	16.7...	14.0...	17.1...	7	11.4...	9	8.052	17.6...	17.4...	15.4...	14.9...	15.9...	18.2...	15.8...	16.3...	13.6...	16.8...	
9	16.1...	13.4...	14.0...	10.8...	14.5...	11.4...	7	9	12.0...	15.4...	14.9...	12.53	11.8...	13.1...	15.8...	13.0...	13.6...	10.2...	14.2...	
10	19.3...	17.1...	17.6...	15.1...	18.0...	4	9.849	8.052	12.0...	18.7...	18.33	16.4...	15.9...	16.9...	19.1...	16.8...	17.3...	14.8...	17.7...	
11	21.6...	19.6...	20.1	17.9...	20.4...	19.1...	16.9...	17.8...	15.4...	18.7...	4	9	9.849	8	21.3...	19.3...	19.8...	17.6...	20.1...	
12	21.26	19.2...	19.6...	17.5...	20.0...	18.7...	16.4...	17.4...	14.9...	18.33	4	9.849	9	8.944	21	18.9...	19.4...	17.2...	19.7...	
13	19.6...	17.4...	17.9...	15.5...	18.3...	16.9...	14.3...	15.4...	12.53	16.4...	9	9.849	4	12.0...	19.3...	17.1...	17.6...	15.1...	18.0...	
14	19.2...	17.0...	17.5...	15.0...	17.9...	16.4...	13.7...	14.9...	11.8...	15.9...	9.849	9	4	12.6...	18.9...	16.7...	17.2...	14.6...	17.6...	
15	20.1	17.9...	18.4...	16.0...	18.8...	17.4...	14.9...	15.9...	13.1...	16.9...	8	8.944	12.0...	12.6...	19.8...	17.6...	18.1...	15.7...	18.5...	
16	21.9...	20	20.4...	18.33	20.7...	19.5...	17.3...	18.2...	15.8...	19.1...	21.3...	21	19.3...	18.9...	19.8...	9	8	12.0...	7	
17	20	17.8...	18.33	15.9...	18.7...	17.3...	14.7...	15.8...	13.0...	16.8...	19.3...	18.9...	17.1...	16.7...	17.6...	9	12.0...	8	11.4...	
18	20.4...	18.33	18.7...	16.4...	19.1...	17.8...	15.3...	16.9...	17.3...	19.8...	19.4...	17.6...	17.2...	18.1...	8	12.0...	9	10.63	11.4...	
19	18.33	15.9...	16.4...	13.82	16.9...	15.3...	12.45	13.6...	10.2...	14.8...	17.6...	17.2...	15.1...	14.6...	15.7...	12.0...	8	9	13.9...	
20	20.7...	18.7...	19.1...	16.9...	19.57	18.2...	15.8...	14.2...	17.7...	20.1...	19.7...	18.0...	17.6...	18.5...	7	11.4...	10.63	13.9...		

Gambar 5.4 Antarmuka Proses MK-NN Sebelum Dilakukan Klasifikasi

DIAGNOSIS PENYAKIT TANAMAN CABAI Menggunakan Metode M-KNN																			
Data Proses MKNN																			
Jumlah Data Latih = 20																			
Jumlah Data Uji = 100																			
nilai K = 5 Proses																			
Euclidean Antar Data Latih Validitas Data Euclidean Data Uji-Data Latih Weighted Voting Hasil Klasifikasi																			
No.	Kelas Data Uji Asli									Kelas Hasil Klasifikasi									
1	BusukBuah									BusukBuah									
2	BusukBuah									BusukBuah									
3	BusukBuah									BusukBuah									
4	BusukBuah									BusukBuah									
5	BusukBuah									BusukBuah									
6	BusukBuah									BusukBuah									
7	BusukBuah									BusukBuah									
8	BusukBuah									BusukBuah									
9	BusukBuah									BusukBuah									
10	BusukBuah									BusukBuah									
11	BusukBuah									BusukBuah									
12	BusukBuah									BusukBuah									
13	BusukBuah									BusukBuah									
14	BusukBuah									BusukBuah									
15	BusukBuah									BusukBuah									
16	BusukBuah									BusukBuah									
17	BusukBuah									BercakDaun									
18	BusukBuah									BusukBuah									
19	BusukBuah									BusukBuah									
20	BusukBuah									BusukBuah									
21	BusukBuah									BusukBuah									
22	BusukBuah									BusukBuah									
23	BusukBuah									BusukBuah									
24	BusukBuah									BercakDaun									

Jumlah Klasifikasi Benar: 94
Jumlah Data Uji: 100
Akurasi: 94.0 %

Gambar 5.5 Tampilan Proses MK-NN Setelah Dilakukan Klasifikasi

BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini menjelaskan tentang pengujian dan analisis dari implementasi MK-NN untuk diagnosis penyakit tanaman cabai. Adapun pengujian yang dilakukan adalah pengujian akurasi sistem.

6.1 Pengujian Akurasi Sistem

Pengujian akurasi sistem dilakukan dengan memasukan nilai k yang bervariasi dengan nilai 5,8,11 dan 14. Jumlah data uji yang digunakan sebanyak 100 data dan jumlah data latih yang digunakan sebanyak 28 data, Data uji ditunjukkan oleh table 6.1 dan data latih ditunjukkan oleh tabel 6.2.

Tabel 6.1 Data Uji 1

G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	Penyakit
4	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?

0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	4	7	9	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	4	7	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	4	0	9	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	7	9	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	6	0	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	6	0	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	6	0	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	6	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	0	8	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	0	8	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	0	8	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	0	8	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	0	0	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	0	0	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	6	8	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	6	8	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	6	8	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	6	8	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	6	0	9	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	6	0	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	8	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	0	8	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	9	0	?

0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	7	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	7	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	7	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	7	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	9	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	?
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	?

Tabel 6.2 Data Latih

Penyakit	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18
BusukBuah	4	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0
BusukBuah	4	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BusukBuah	4	6	7	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
BusukBuah	4	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BusukBuah	4	6	0	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	7	9	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	7	0	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	0	9	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	0	7	9	0	0	0	0	0	0	0	0	0	0
LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	4

LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	0
LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	4
LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	0	0	0	0	0	0	0
LayuFusarium	0	0	0	0	0	0	0	0	5	6	0	9	0	0	0	0	0	4
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	9	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	0	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	9	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	0	0	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	8	9	0
BusukBuah	4	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0
BusukBuah	4	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	7	9	0	0	0	0	0	0	0	0	0	0
BercakDaun	0	0	0	0	0	4	7	0	0	0	0	0	0	0	0	0	0	0
LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	4
LayuFusarium	0	0	0	0	0	0	0	0	5	6	8	9	0	0	0	0	0	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	9	0
VirusGemini	0	0	0	0	0	0	0	0	0	0	0	0	4	5	7	8	0	0

6.1.1 Pengujian dengan nilai K = 5

Pengujian yang pertama adalah pengujian dengan nilai K = 5, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 5 ditunjukkan oleh tabel 6.3.

Tabel 6.3 Hasil Uji K = 5

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BusukBuah	BercakDaun
17	BusukBuah	BusukBuah

18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BusukBuah	BercakDaun
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BusukBuah	BusukBuah
29	BusukBuah	BusukBuah
30	BusukBuah	BusukBuah
31	BusukBuah	BusukBuah
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium
41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	LayuFusarium	LayuFusarium
54	LayuFusarium	BercakDaun
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium
59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium

61	LayuFusarium	LayuFusarium
62	LayuFusarium	LayuFusarium
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	LayuFusarium	BercakDaun
70	VirusGemini	VirusGemini
71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini
80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini
84	VirusGemini	VirusGemini
85	VirusGemini	BercakDaun
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	VirusGemini	BercakDaun
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	VirusGemini
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 6 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasiya menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut:

$$\begin{aligned} Akurasi &= \frac{94}{100} \times 100\% \\ &= 94\% \end{aligned}$$

6.1.2 Pengujian dengan nilai K = 8

Pengujian yang pertama adalah pengujian dengan nilai K = 8, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 8 ditunjukkan oleh tabel 6.4.

Tabel 6.4 Hasil Uji K = 8

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BusukBuah	BercakDaun
17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BusukBuah	BercakDaun
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah

28	BusukBuah	BusukBuah
29	BusukBuah	BusukBuah
30	BusukBuah	BusukBuah
31	BusukBuah	BusukBuah
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium
41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	LayuFusarium	BercakDaun
54	LayuFusarium	BercakDaun
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium
59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium
61	LayuFusarium	LayuFusarium
62	LayuFusarium	BercakDaun
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	LayuFusarium	BercakDaun
70	VirusGemini	VirusGemini

71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini
80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini
84	VirusGemini	VirusGemini
85	VirusGemini	BercakDaun
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	VirusGemini	BercakDaun
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	VirusGemini
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 8 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasiya menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned}
 Akurasi &= \frac{92}{100} \times 100\% \\
 &= 92\%
 \end{aligned}$$

6.1.3 Pengujian dengan nilai K = 11

Pengujian yang pertama adalah pengujian dengan nilai K = 11, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 11 ditunjukkan oleh tabel 6.5.

Tabel 6.5 Data Uji K = 11

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BusukBuah	BercakDaun
17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BusukBuah	BercakDaun
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BusukBuah	BusukBuah
29	BusukBuah	BusukBuah
30	BusukBuah	BercakDaun
31	BusukBuah	BercakDaun
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun

37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium
41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	LayuFusarium	BercakDaun
54	LayuFusarium	BercakDaun
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium
59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium
61	LayuFusarium	BercakDaun
62	LayuFusarium	BercakDaun
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	LayuFusarium	BercakDaun
70	VirusGemini	VirusGemini
71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini

80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini
84	VirusGemini	VirusGemini
85	VirusGemini	BercakDaun
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	VirusGemini	BercakDaun
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	BercakDaun
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 12 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasi menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{88}{100} \times 100\% \\
 &= 88\%
 \end{aligned}$$

6.1.4 Pengujian dengan nilai K = 14

Pengujian yang pertama adalah pengujian dengan nilai K = 14, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 14 ditunjukkan oleh tabel 6.6.

Tabel 6.6 Hasil Uji K = 14

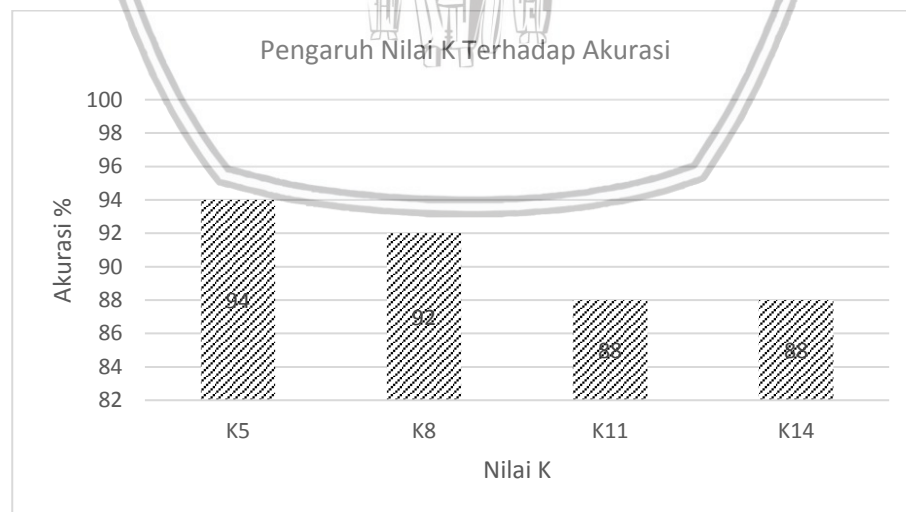
No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BusukBuah	BercakDaun
17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BusukBuah	BercakDaun
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BusukBuah	BusukBuah
29	BusukBuah	BusukBuah
30	BusukBuah	BercakDaun
31	BusukBuah	BercakDaun
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium

41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	LayuFusarium	BercakDaun
54	LayuFusarium	BercakDaun
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium
59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium
61	LayuFusarium	BercakDaun
62	LayuFusarium	BercakDaun
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	LayuFusarium	BercakDaun
70	VirusGemini	VirusGemini
71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini
80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini

84	VirusGemini	VirusGemini
85	VirusGemini	BercakDaun
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	VirusGemini	BercakDaun
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	BercakDaun
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 12 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasiya menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$Akurasi = \frac{88}{100} \times 100\% \\ = 88\%$$



Gambar 6.1 Grafik Hasil Pengujian Nilai K

6.2 Analisis

Setelah dilakukan pengujian pengaruh nilai K terhadap keakuratan hasil, maka kemudian akan dianalisis hasil dari setiap pengujian, dari pengujian yang dilakukan sebanyak empat kali didapatkan hasil pengujian yaitu, pengujian pertama pada dengan nilai $K=5$ didapatkan akurasi yang tinggi dengan nilai 94%, sedangkan untuk pengujian kedua dengan nilai $K = 8$ akurasi mulai turun menjadi 92%, ketika dilakukan pengujian ketiga dengan nilai $K = 11$ didapatkan akurasi yang lebih rendah lagi yaitu dengan nilai akurasi sebesar 88% dan ketika dilakukan pengujian yang terakhir dengan nilai $K = 14$ nilai akurasi sama dengan pengujian ketika $K = 11$, nilai akurasi yang dihasilkan yaitu 88%. Dari keseluruhan hasil pengujian diketahui bahwa ketika nilai K semakin tinggi maka terjadi kecenderungan akurasinya semakin rendah.



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan pengujian yang telah dilakukan terhadap diagnosis penyakit tanaman cabai menggunakan *Modified K-Nearest Neighbor* (MKNN) adalah sebagai berikut:

1. Implementasi algoritma *Modified K-Nearest Neighbor* (MKNN) untuk mengklasifikasikan penyakit tanaman cabai berdasarkan 18 gejala penyakit dapat memberikan diagnosa awal terhadap 4 jenis penyakit cabai yaitu, penyakit busuh buah, bercak daun, layu fusarium dan penyakit virus.
2. Hasil klasifikasi diambil berdasarkan nilai akurasi tertinggi yang didapatkan dari hasil pengujian adalah ketika $K = 5$ yaitu dengan nilai 94%.
3. Nilai K mempengaruhi tingkat akurasi sistem dimana semakin tinggi nilai K maka terjadi kecenderungan akurasi akan semakin turun.

Berdasar pada hasil uji dapat didapat kesimpulan algoritma metode *Modified K-Nearest Neighbor* (MKNN) menunjukkan akurasi yang baik melakukan klasifikasi penyakit cabai

7.2 Saran

Saran yang dapat diberikan dari penelitian ini untuk penelitian selanjutnya, yaitu:

1. Menambah kategori gejala penyakit cabai dengan gejala lain.
2. Menggunakan data latih dengan komposisi kelas penyakit yang lebih seimbang dan menggunakan nilai k yang lebih optimal agar didapatkan nilai akurasi yang lebih baik.
3. Mengembangkan dan/atau melakukan kombinasi metode *Modified K-Nearest Neighbor* (MKNN) dengan metode lain untuk hasil akurasi yang lebih optimal.

DAFTAR PUSTAKA

- Biro Pusat Statistik. 1994. Survei Pertanian Produksi Tanaman Sayuran dan Buah-buahan di Indonesia. Jakarta: BPS.
- Biro Pusat Statistik. 1995. Survei Pertanian Produksi Tanaman Sayuran dan Buah-buahan Semusim di Jawa. Jakarta: BPS.
- Biro Pusat Statistik. 1998. Statistik Perdagangan Luar Negeri Indonesia. Ekspor. Jilid I & II. Jakarta: BPS.
- Djarwaningsih, T. 1983. Pemanfaatan jenis-jenis cabai (*Capsicum*spp.) sebagai tanaman hias. *Buletin Kebun Raya*6 (2): 45-52.
- Djarwaningsih, T. 1986. Jenis-jenis *Capsicum*L. (*Solanaceae*) di Indonesia. *Berita Biologi*3 (5): 225-228.
- Djarwaningsih, T. 1990. Cabai merah dan kerabatnya di Indonesia. *Nekabija*01: 18-21.
- Eshbaugh, W.H. 1970. A Biosystematic and evolutionary study of *Capsicum baccatum*(*Solanaceae*). *Brittonia*22: 31-43.
- Heiser, C.B. 1969a. *Nightshades*. San Francisco: Freeman.
- Khotimah, H., 2015, Penentuan Status Gizi Balita menggunakan Metode Modified K-Nearest Neighbor (MK-NN) (Studi Kasus: Kecamatan Kertosono), Universitas Brawijaya, Malang.
- Parvin H., Hoseinali., & Behrouz M. 2010. Modification on K-Nearest Neighbor Classification. *Global Journal of Computer Science and Technology* Vol.10 Issue 14 (Ver.1.0).
- Putri, M.B.P., 2017. Diagnosis Penyakit pada Kucing Menggunakan Metode Modified K-Nearest Neighbor, Universitas Brawijaya, Malang.
- Suwarna, A., M. Sigit, dan M. Santoso. 2006. Pertumbuhan dan Hasil Cabe Merah (*Capsicum annum*) pada Andisol yang Diberi Mikoriza, Pupuk Fosfor, dan Zat Pengatur Tumbuh
- Wafiyah, F., Hidayat, N., Perdana, R.S., 2017, Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JPTIIK)*, 1(10), p.1210-1219
- Zulaikha, S., dan Gunawan. 2006. Serapan fosfat dan respon fisiologis tanaman cabai merah cultivar hot beauty terhadap mikoriza dan pupuk fosfat pada tanah ultisol. *Bioscientiae*, 3(2): 83-92.